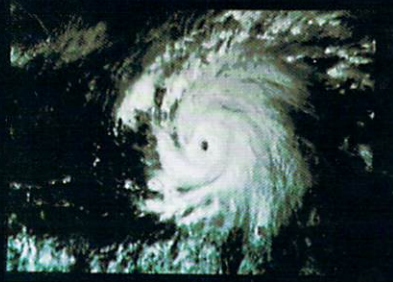
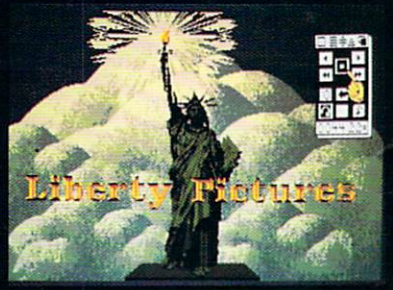
All About Amiga Video

Amazing Computing™

Volume 2 Number 7
U.S.A. $3.50
Canada $4.50

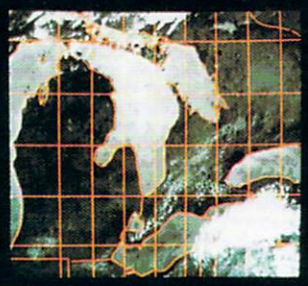
Commodore Amiga™ Information & Programs





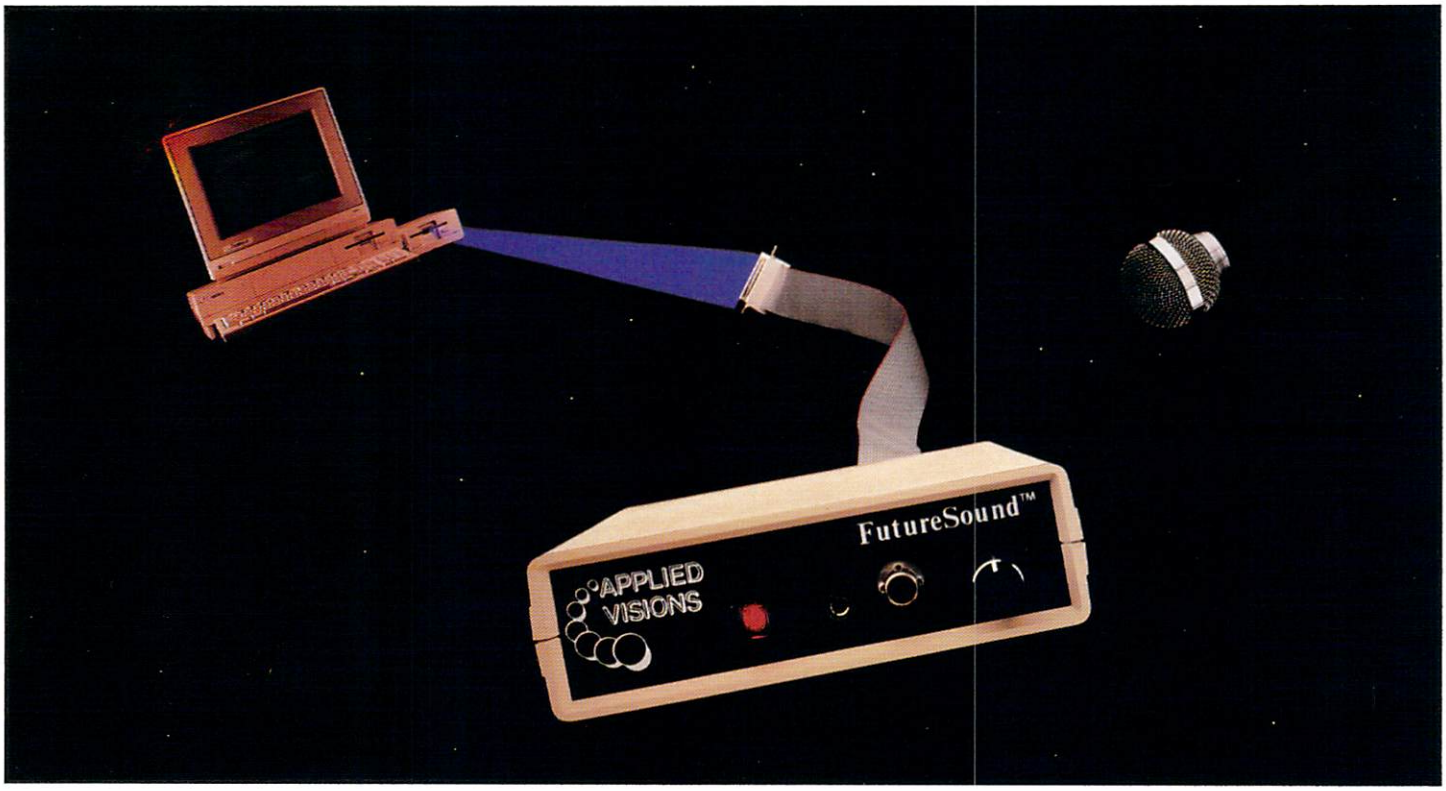
AMAZING VIDEOS

AMIGA VICE



AMIGA

An Inside Look at the CSA Turbo-Amiga!



"Open the pod bay doors, HAL..."

Programmers cast their vote!

Right now, leading software developers are hard at work on the next generation of Amiga® products. To add the spectacular sound effects we've all come to expect from Amiga software, they are overwhelmingly choosing one sound recording package... FutureSound. As one developer put it, "FutureSound should be standard equipment for the Amiga."

FutureSound the clear winner...

Why has FutureSound become the clear choice for digital sound sampling on the Amiga? The reason is obvious: a hardware design that has left nothing out. FutureSound includes two input sources, each with its own amplifier, one for a microphone and one for direct recording; input volume control; high speed 8-bit parallel interface, complete with an additional printer port; extra filters that take care of everything from background hiss to interference from the monitor; and of course, a microphone so that you can begin recording immediately.

What about software?

FutureSound transforms your Amiga into a powerful, multi-track recording studio. Of course, this innovative software package provides you with all the basic recording features you expect. But with FutureSound, this is just the beginning. A forty-page manual will guide you through such features as variable sampling rates, visual editing, mixing, special effects generation, and more. A major software publisher is soon to release a simulation with an engine roar that will rattle your teeth. This incredible reverberation effect was designed with FutureSound's software.

APPLIED VISIONS

Question: What can a 300 pound space creature do with these sounds?

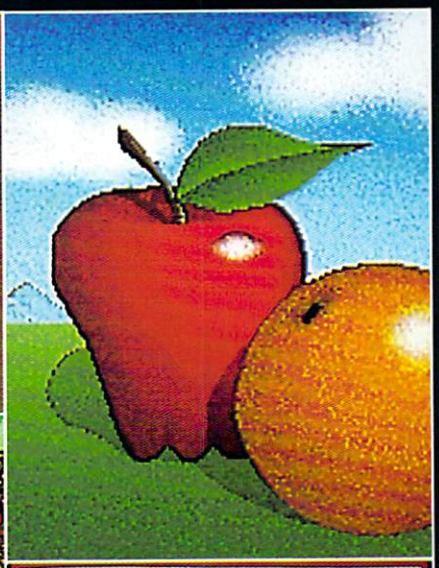
Answer: Anything he wants.

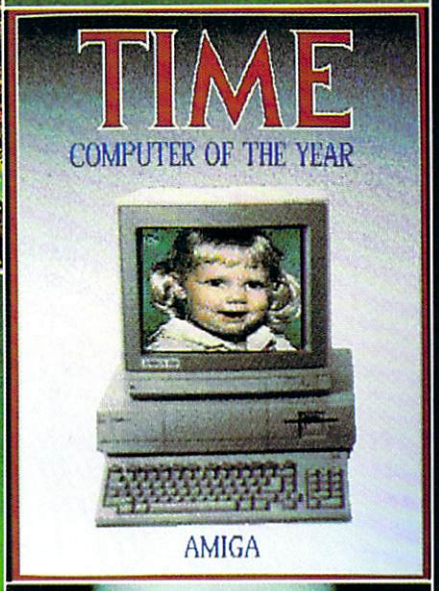
Since FutureSound is IFF compatible (actually three separate formats are supported) your sounds can be used by most Amiga sound applications. With FutureSound and Deluxe Video Construction Set from Electronic Arts, your video creations can use the voice of Mr. Spock, your mother-in-law, or a disturbed super computer.

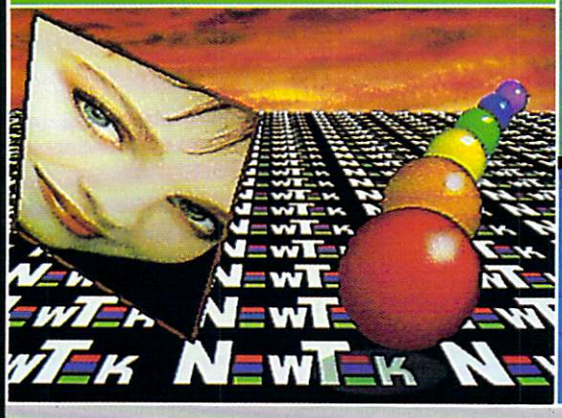
Programming support is also provided. Whether you're a "C" programming wiz or a Sunday afternoon BASIC hacker, all the routines you need are on the non-copy protected diskette.

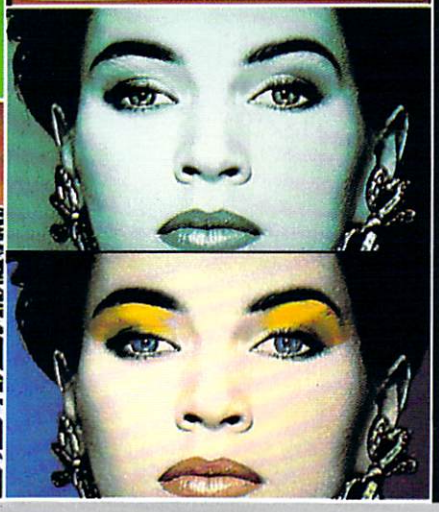
Your Amiga dealer should have FutureSound in stock. If not, just give us a call and for $175 (VISA, MasterCard or COD) we'll send one right out to you. Ahead warp factor one!

Applied Visions, Inc., Suite 2200, One Kendall Square
Cambridge, MA 02139 (617) 494-5417









ONLY DIGI PAINT CAN DO ALL THIS

Get the maximum graphics power from your Amiga. Create stunning, lifelike computer artwork with Digi-Paint, the first full-featured 4096 color (Hold and Modify) paint program. Break the "32 color barrier" and finally realize the potential of your Amiga with Digi-Paint's advanced features:

- 4096 colors on screen simultaneously
- NewTek's exclusive enhanced HAM mode
- Dithered HAM gradient fill
- Full screen effects including double, half size, mirror reverse and more
- Full IFF and Digi-View compatibility
- Use 320x200 or HAM hi-res 320x400 resolutions
- Fat bits Magnify mode
- Rectangle, oval, line and other drawing tools
- 12 different paint modes including blending, tinting and smooth shading
- Full lasso cut and paste with automatic edge blending
- Programmed completely in assembly language for fast, smooth response

Find out why Byte Magazine called Digi-Paint "Remarkable". Available now at your local Amiga dealer or call: **1-800-843-8934.**

ONLY $59.95

NEWTEK
INCORPORATED

An Inventive Approach To Practical Reality

ZING! T.M.

Sometimes you just have to take matters into your own hands. Or hand. More than once, a brainstorm has turned into a shower of practical reality. To the skeptic, the truth can be shocking.

Well, now you can start your AMIGA flying high with the dynamic, new ZING! utility programs from MERIDIAN SOFTWARE, INC.

ZING! programs are the best program companions for the AMIGA on the market today. Separately, they are practical, yet dynamic working partners. Strung together. . .the sky's the limit!

The High Point In Software Innovations. . .

MERIDIAN™ SOFTWARE INC.

P.O. Box 890408 Houston, Texas 77289-0408
(713) 488-2144

ZING! The fastest and easiest way to work with the AMIGA. ZING! is actually a collection of utility programs which, after installation, are invisible until called up by the user through the use of hot keys. The main goal of ZING!, is to essentially eliminate the need to learn the operating system protocol and cryptic commands. Functions include: copying, editing, deleting, sorting, renaming, searching, reorganizing files, and much more! Shipping now for: $79.95

ZING!Keys ZING!Keys is a sophisticated, reprogrammable MACRO and Hot Key program. A program which can stand on its own, or be tied to ZING! You can train ZING!Keys to accomplish the most annoyingly repetitive tasks in a much easier fashion. Plus, ZING!Keys allows you to retrieve commands and reuse them, as well as the ability to record mouse movements and use them with the press of a key. Shipping now for: $49.95

ZING!Spell As the name implies, ZING!Spell is a program which can detect misspelled words as you are typing them. A simple interface allows you to correct a word, or add a word to the dictionary. You'll never need to buy another spelling checker. Can be used with other ZING! components or alone. $49.95

Please add $3.00 shipping and handling to each order.

Credit cards and dealer inquiries welcome.

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™.
They carry Amazing Computing™, your resource for information on the Amiga™.

If you are not an Amazing Dealer, but would like to become one, contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA. 02722
1-617-678-4200

Alabama
Abax Data Systems — Huntsville
Alabama Book Store — Tuscaloosa
Command Computers — Birmingham
Electronics Boutique — Hoover
Madison Books & Computers — Madison
Mail Tel Communications — Mobile
Mel's Photo & Computer Shop — Montgomery
Universal Computer System — Mobile

Arkansas
SIS Inc. — Mt. Home
The Micro Shop — Little Rock

Arizona
Computer West — Phoenix
Computer Works — Glendale
Copperstate Business Systems — Phoenix
Copperstone Computers — Tucson
Hacko's — Tempe
North American Digital — Tucson

California
Aday International — San Leandro
Augnet — Freemont
Balboa Office Supply — San Diego
Bits n Pieces Comp Prod. — Anahiem
Brown Knows Computing — Redlands
Byte Factory Computers — Salinas
C.F.T. — Clairemont
Century Computer Systems — La Habra
Circus of Books II — LosAngeles
Commotech — National City
Compare Inc. — Ridgecrest
Computer Attic — Palo Alto
Computer Nook, Inc. — San Bernadino
Computer Outlet — San Diego
Computer Outlet — San Marcus
Computer Time — Citrus Hts
ComputerLand Freemont — Freemont
Dataphile — Watsonville
Delta Spectrum — Laguna Beach
De Lauer's News Agency — Oakland
Electronics Boutique — West Los Angeles
Empire Computers — Santa Rosa
Harding Way News — Stockton
Home Computing Center — Concord
Home Computing Center — Mill Valley
Home Computing Center — San Leandro
Home Computing Center — San Bruno
HT Electronics — Sunnyvale
K J Computers — Granada Hills
Levity — North Hollywood
LTC Computers — Santa Ana
Micro Flight — San Jacinto
Nonagon Software — El Cerrito
Op Amp Tech Books — Los Angeles
World Book & News — Hoolywood
R&R Software — Chula Vista
R&R Software — San Diego
Ridgecrest Computer Ctr. Inc. — Ridgecrest
S.O.S. Computers — Los Angeles
Software First — Santa Rosa
Software Cellar — Anahiem
Software Plus — Citrus Hieghts
Tower Books — Chico
Tower Books — Citrus hights
Tower Books — Concord
Tower Books — Mountain View
Tower Books — Sacramento
Tower Books — San Mateo
The Computer Room — Scotts Valley
Winner's Circle — Berkley

Colorado
Citadel Computers — Colorado Springs
Computer Discount — Denver
Computer Room — Aurora
Ebert Personal Computers — Aurora
Micro Computer World — Fort Collins
Micro World — Lakewood
Sun Country Computers — Colorado Springs
Aurora Newsstand — Aurora
Whole Life Distributors — Englewood

Connecticut
Amiga Business Computer — Norwalk
Computer Outlet — PlaVernon
Electronics Boutique — Danbury
Friendly Computers — Plantsville
Mnemonics Inc. — Norwalk
Personal Computer center — Norwich
Software City — Orange
Software Kingdom — East Windsor
Spectrum Computers — New Haven

Delaware
Castle Video — Newark

Florida
A A Computers — Jacksonville
Book Mania — Orlando
Commodore Country Inc. — Pinellas Park
Computer Bar — Pennsacola
Computer Base — Brandon
Computer Image — Miami
Computer Specialty — Melbourne
Computer Terminal Inc. — Sebring
Computers 4 Rent — W. Palm Beach
Cotronics Inc. — Stewart
Education Computers Etc. — Talahasse
Electronics Boutique — Boynton Beach
Electronics Boutique — Tallahassee
Electronics Boutique — Miami
Electronics Boutique — Orlando
Family Computers — Sarasota
Florida Book Store — Gainsville
Games 'N' Gadgets — Orange Park
Games 'N' Gadgets — Jacksonville
Games 'N' Gadgets — Tallahassee
Games 'N' Gadgets — Clearwater
Gulf Coast Computer Exchange — Panama City
Heartland Computers — Mulberry
K&L Computers — E. Palatka
Megaport Computer Center — South Venice
Micro Computer Center — Largo
Micro's Etc. — Altamonte Springs
New Age Electronics — St. Petersburg
Random Access Computers Inc. — Ft. Walton Beach
The Open Door — Cocoa
The PC Collection — Coral Gables

Georgia
Electronics Boutique — Atlanta
Electronics Boutique — Atlanta
Electronics Boutique — Kennesaw
Future System — Augusta
Games 'N' Gadgets — Atlanta
Games 'N' Gadgets — Atlanta
Software South Inc. — Savannah
The 64 Store — Atlanta

Hawaii
Astro Computers & Electronics — Aiea
Computer House Inc. — Honolulu
J & C Repair — Aiea

Idaho
A.B.I. Computer & Video — Boise
A.B.I. Computer & Video — Nampa

Illinois
A Bicketer — Rockford
Computerland O'Hare — Park Ridge
ComputerLand of Niles — Niles
DigitalWorld — Addison
I.C.V. Data Center Inc. — Mundelein
Ilini Microcomputers Inc. — Naperville
Interstate Computer Service — LeMont
Krach's & Brentano's — Chicago
Micro Age — Oaklawn
Software Or Systems — Fairview Hts.
Software Plus — Wheeling
Software Plus West — Hanover Park
The Memory Expansion — Normal
Unique Computer — Sterling

Indiana
Burkat Computer Ctr. — South Bend
Bytrex Computer — Ft. Wayne
Bytrex of Angola — Angola
Computer Corner — Marion
Computer Corner — Fort Wayne
Computer People Inc. — Michigan City
Greg Cheney — Greenfield
Micro Computer Centers Inc. — Indianapolis
Spectra Computer Sys. Inc. — Evansville
Vons Computers — West Lafayette

Iowa
Century Systems — Des Moine
Gronet Computers — Des Moine
Micro Computer Applications LTD — Marshaltown

Kansas
J & L Electronics — Liberal
MidKansas Computers — Newton
Midwest Computers — Manhattan
Thorobred Computers — Topeka

Kentucky
Computer Dimensions Inc. — Lexington
Reality World Computers — Lexington
The Software Source — Louisville

Louisiana
Computer Time — Shreveport
Electronics Boutique — Kenner
Electronics Boutique — Monroe
Electronics Boutique — Lawrenceville
Modern Business Machines — Metairie
Software Center International — Metairie
Software Solutions — Baton Rouge
Software Source — Lafayette
The Computer Clinic — Lafayette

Maryland
Buried Treasure — Rockville
Computer Crafters — Wheaton
Computerware TODAY — Gaithersburg
Compuvision Vision Center — Columbia
Electronics Boutique — Baltimore
Electronics Boutique — Baltimore
Electronics Boutique — Anne Arundel County
Electronics Boutique — Owings Mills
Games 'N' Gadgets — Baltimore
Games 'N' Gadgets — Baltimore
Games 'N' Gadgets — Columbia
Layhill Newsstand — Kensington
Micro Computer Center — Baltimore
Software Advantage — Rockville
Software N Things — College Park
Waldorf Computer — Waldorf

Massachusetts
Club Computer — Cambridge
E.U. Wurlitzer Co. — Boston
Electronics Boutique — Holyoke
General Computer Store — Framingham
HCS Computer Center — Marshfield
LCA Video & Computer Ctr. — Norwood
Memory Location — Wellsley
OmniTek Computers — Tewksbury
Pioneer Valley Data Equipment — Amherst
The Bit Bucket — West Newton
The Software Haus — Quincy
Tycom Inc. — Pittsfield

Maine
Computer Barn — Damariscotta

Michigan
Basic Business Control Sys. — Traverse City
Basic Computer Center — Grand Rapids
Bits-Bytes-Nibbles — Petowskey
Canton Computer — Canton
CompuSoft — Kalamazoo
Direct Access — Lansing
Edmunton Computer Center — Pontiac
Galaxy Computers — Allen Park
Global Computer Center — Dearborn
I O Software — Flint
Michigan Software — Novi
Midland Computer Shop — Midland
Central Park Computers — Okemos
Pro-Video — Okemos
Programs Unlimited — Grand Rapids
Roseville Computer Store — Warren
Slipped Disk — Madison Heights
Slipped Disk — Southfield
Software Plus — Grand Rapids
Specialists Computer — Livonia
State Street Computers — Ann Arbor
T.J. Enterprises — Clinton
The Software House — Kalamazoo
Ye Olde Computer Shoppe — Ypsilanti

Minnesota
Computers, etc. — Eagan
Computer Outfitters — Roseville
Computer Place — Minneapolis
JMH Software — Maple Grove
MCD of Hibbing Inc. — Hibbing
Midwest Computer Services — Clearwater
OnLine Computers — Minnetonka
Specialists IN — Hopkins
Valiant Inc. — Stillwater

Mississippi
Electronics Boutique — Ridgeland
Enterprises Unlimited — Jackson
National Computer Center — Ocean Springs

Missouri
Associated Computer Services — Springfield
Brands Mart Computers — Kansas City
Data Byte Computer Co. — St. Louis
Instant Replay Ltd. — Chesterfield
Systems Plus — Independence
Systems Plus of St. Louis — St. Louis

Montana
Appelgren Computer Systems — Great Falls

Nebraska
BULLS EYE — Lincoln
My Com Inc. — Plattsmouth
Double E Electronics — Omaha

Nevada
Century 23 Inc. — Las Vegas
Computer World — Las Vegas
ComputerLand of Carson — Carson City
Tower Records — Las Vegas

New Hampshire
Computer Mart of New Hampshire — Nashua
Diversified Computers — Keene
Electronics Boutique — Nashua

New Jersey
Electronics Boutique — Voohrees
Electronics Boutique — Wayne
Electronics Boutique — Rockaway
Electronics Boutique — Cherry Hill
Electronics Boutique — Woodbridge
Family Computer Centres — South Orange
Games 'N' Gadgets — Burlington
Games 'N' Gadgets — Livingston
Morris County Stationers — Chester

New Mexico
Academy Computers — Albuquerque
New Horizon Computer System — Alamogordo
Page One Newsstand — Albuquerque
Technological Concepts — Las Cruces

New York
Amuse — New York
Amicus- Hudson Valley — Wappingers Falls
Amiga Business Computer — North Port
Byte Shop — Merrick
CIA Software Center — Flushing
Compurite Center — White Plains
Computer Outlet — Jamestown
Computer Palace — Felden
Cromland Inc. — Brooklyn
Electronics Boutique — Lake Grove
Electronics Boutique — Huntington Station
Electronics Boutique — Albany
Electronics Boutique — Valley Stream
Games 'N' Gadgets — Nanuet
Games 'N' Gadgets — Garden City
Leigh's Computers — New York
Mnemonics — North Port
Software Centre International — Forest Hills
Software & Such — Scotia
Software City — Forest Hills
Software Supermarket — Kenmore
Star Tech Systems — Massena
Syntiste Trading — New York
Tadiran — New York
Tedrow Business Products — Rochester
The Computer Cellar — Albany
The Working Computer — Stoney Brook
Video Computer Bin — Brooklyn
Video Computer Center — Rome
Video Connection — Schenectady
Village Computer Inc. — New York
World Computers Inc. — Hicksville

North Carolina
Digitz — Raleigh
Games 'N' Gadgets — Raleigh
Software City — Raleigh
Software City — Charlotte
TDS Computers — Durham
Triad Computers — Greensboro

North Dakota
Computer Associates — Fargo
G. F. Computers — Grand Forks
The Computer Store — Minot

Ohio
Computer Network — Newark
Computer Site — Cleveland
Earthrise Micro Systems — Columbus
Electronics Boutique — Columbus
Fairborn Home Computer — Fairborn
Games 'N' Gadgets — North Randall
Lakes Consumer Electronics — Akron
Main Office Supply — Coshocton
Micro Center — Columbus
Microwave Magic — Fairfield
North Coast Programming — Willoughby
Personal Computer Co. — Youngstown
Program Store — Columbus
Quality Computer Appliances — Toledo
Stanlan Computer Center — Sandusky
Software Centre International — North Olmsted
Software and More — Cincinnati
Software City — Centerville
Software Connection — Mayfield Hts.

Oklahoma
Colonial Video & Computer — Bardesville
Micro Link Computer Store — Tulsa
P C Tech — Tulsa
Second Hand Software — Oklahoma City
Video-Comp Inc. — Lawton

Oregon
Andromeda — LaGrande
BTE Telephone & Computer — Klamath Falls
Clackamas Computers — Beaverton
Clackamas Computers — Clackamas
Computron Business Systems — Portland
Software Express — Eugene

Pennsylvania
Alpha Omega EBE — Warren
Basic Computer Systems — Hermitage
Circle Computer — Ephrata
Computers and Games — Reading
Data Sotique Computerware — Pittsburg
Downington Computer Center — Downington
East Coast Software — Hershey
Electronics Boutique — North Wales
Electronics Boutique — Langehorne
Electronics Boutique — Exton
Electronics Boutique — Norristown
Electronics Boutique — Pittsburgh
Electronics Boutique — King of Prussia
Electronics Boutique — Whitehall
Games 'N' Gadgets — Lancaster
Games 'N' Gadgets — King of Prussia
Games 'N' Gadgets — Media
Games 'N' Gadgets — West Mifflin
Gene's Books — King of Prussia
North Central Digital Systems — Danville
Pittsburgh Computer Store — Pittsburgh
The Electronic Boutique — Norristown
Triangle Computers — Indiana

Rhode Island
International Computer Services — Johnston
Micro Limits — Smithfield
Micro Limits — Warwick

South Carolina
Avcom International Inc. — West Columbia
Electronics Boutique — Greenville
Horizon — Greenville
Micro Computer Depot — Sumter
Watson's Jewlery — Chesterfield

South Dakota
Computer Business Center — Rapid City

Tennessee
ComputerLab of Memphis — Memphis
Games 'N' Gadgets — Memphis
Software First — Nashville

Texas
Byte Shop — Austin
C.E.M. Corporation — Brazoria
Colonial Video — Houston
Computer Age — Houston
Computer Age — Dallas
Computer Magic — Austin
Computer Revelations — Houston
Computer Time Inc. — Marshall
FNS Computer Services — Texarkana
Metropolitan Computer — Richardson
Micro Search — Houston
Softex — Watauga
Software Library — Wichita Falls
Software Plus — El Paso
Software Terminal — Fort Worth
The Computer Experience — San Antonio
The Computer Stop — Abilene

Utah
Computers Plus — Murray

Vermont
Ray Supply Inc. — South Burlington

Virginia
Colony Ltd. — Hampton
Diskovery — Falls Church
Family Computer Center — Fairfax
Games 'N' Gadgets — Hampton
Games 'N' Gadgets — Virginia Beach
Games 'N' Gadgets — Virginia Beach
Software Central — Norfolk
Software City — Richmond
Virginia Micro Systems — Woodbridge

Washington
A.P.P.L.E. CO-OP — Renton
Apple Valley Computers — E. Wenatchee
Bull Dog News — Seattle
Byte World — Kent
Com-Soft Computers — Everett
Computers + Inc. — Tacoma
Computer Outlet — Edmonds
Family Computers — Lynwood
Kitsap Amiga Operator System — Bellview
Micro Age Computer Store — Bellview
Micro Specs Inc. — Spokane
Nibble Shi — Tacoma
Omni International — Seattle
Programs Plus — Seattle
Take-A-Byte — Kennewick

West Virginia
Layhill Newsstand — Charleston

Wisconsin
Celotron Computers — Racine
Computer Software Center — Milwaukee
Fox Valley Personal Computer — Oshkosh
Mom's Computers — Stevens Point
Software Shoppe — Fond Du Lac
TMW Software — Greenbay
TMW Software Inc. — Wausau

Wyoming
Byte Emporium — Laramie

Canada
Compute or Play — Alberta
Computerworks — Alberta
Computer Shop — Alberta
Data Pro — Alberta
Computer Shop of Calgary — Alberta
The Software House — Alberta
Hindsin Computing — Alberta
Red Deer Micro Systems — Alberta
Memory Land — Alberta
Computer King — British Columbia
Cord — British Columbia
Nutec — British Columbia
Sprita — British Columbia
Strider — British Columbia
Super Software — British Columbia
Canadian Computer Era — Manitoba
Micromart — Manitoba
ComputerWorld — Manitoba
J.W. Allen — Manitoba
Kobetak — Manitoba
Minicomp — Manitoba
Arkon — Ontario
Chatam Computer — Ontario
Computerland — Ontario
Compec — Ontario
Electronics 2001 — Ontario
Hastings Data — Ontario
The Computer & You — Ontario
Thornhill — Ontario
Software Co. — Quebec
Dettronic Solare — Quebec
Merodata — Quebec
Info Mereler — Quebec
Laval Micro — Quebec
Vision Tronique — Quebec
Distribution Vistpro — Quebec

Distributed in Canada by
Avita Software Distributors — Concord, Ontario
Compulit Distributors — British Columbia
Phase 4 Dist. Inc. — Calgary, Alberta

Argentina
Logic Computer — Buenos Aires

Australia
High Technology Computer Supplies — Brighton, Melborn
Maxwell Office Equipment — Abbotsford, Melburn

Japan
Pineapple 6502 — Tokyo, 111

Saudia Arabia
Abdulla Fouad & Sons — Daman

Switzerland
Software Land AG — Zuerich

Amazing Computing™ is also available in most B. Dalton Booksellers stores, B. Dalton Software stores, and Software Etc. locations.

Amazing Computing™

Publisher: Joyce Hicks
Circulation Manager: Doris Gamble
Assistant to the Publisher: Robert James Hicks
Traffic Manager: Robert Gamble
Managing Editor: Don Hicks
Submissions Editor: Ernest P. Viveiros Jr.
Hardware Editor: Ernest P. Viveiros Sr.
Amicus & Technical Editor: John Foust
Music & Sound Editor: Richard Rae
Art Director: Keith M. Conforti
Advertising Manager: John D. Fastino
Copy Editor: Michael T. Cabral
Production Manager: Mark Thibault
Assistant P.M.: Keven Desmarais

Advertising Sales & Editorial
1-617-678-4200

Special thanks to John at J & B Photo and Paul Boden at Software Supermarket.

Amazing Computing™ (ISSN 0886-9480) is published by PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722. Subscriptions: in the U.S. 12 issues for $24.00; in Canada & Mexico, $30.00; Overseas; $35.00. Printed in the U.S.A.

First Class or Air Mail rates available upon request.

PiM Publications, Inc. maintains the right to refuse any advertising.

PiM Publications, Inc. is not obligated to return unsolicited materials. All materials requesting return must be received with a Self Addressed Stamped Mailer.

Amazing Contents

Volume 2, Number 7

Amazing Video...

Amazing Programming...

Amazing Reviews...

Amazing Columns...

Amazing Departments:

Amazing Mail:

Dear Repartee:
A lot of us here in AmigaLand are tempted to use 3 1/2 single-sided disks because of their price, and use them as double-sided disks.

Some users, however, contend that, unlike earlier Commodore Computers which used 5 1/4 inch diskettes that could often be used as doubled-sided without damage to the disk drive itself, there is an inherent danger in using single-sided disks on the Amiga.

You would do a lot of us a favor and clear up a point of argument if you would publish a response to this question in your excellent magazine!

Sincerely:
Robert V. Night
West Des Moines, IA

Sorry, different disk manufacturers are reportedly producing 3 1/2 disk media in different ways. While some manufacturers produce both single-sided diskettes and double-sided diskettes in the same manner and then test the second side of the media for double-sided certification, other manufacturers begin by burnishing only one side of the single-sided media and two sides of the double-sided media.

Burnishing produces a nice, clean image for smooth disk operation and better data storage. This added process means less wear on the disk drive head. The result is fewer failures of media and hardware.

However, we are extremely fortunate. The 3 1/2 double-sided disk is rapidly receiving approval as the new standard. This acceptance has caused a good many disk manufacturers to jump into the market. The resulting competition has driven down the market price of 3 1/2 media. It is questionable whether the current price for media will remain so low, with the present trade tension between Japan and the United States.

However, the difference in price between double-sided and single-sided 3 1/2 media is insignificant . . . especially when compared to the risk of data or hardware failure.

Dear Sirs,
I have read the BBS listings in your magazine and I would like to have my BBS listed.

Here, then, is the information on my BBS:

BBS Name: Portal BBS
Sysop : Charles Boesel(Boesel)
Phone # : 314-867-9041
Hours : 24 hrs a day, 7 days a week
Modem : 300-1200 Baud
City : St. Louis, Missouri
Files : Amiga and IBM software; Amiga artwork and 10 message areas In 20 MEG of On-Line Storage

Thank You,
Charles Boesel
St. Louis, MO

Good Luck on your BBS. Support is always welcomed and encouraged on the Amiga. With the adition of Sidecar and the Amiga 2000 Bridgecard, the IBM area will be a great bonus.

Dear Amazing Computing,
I'm writng to you on behalf of the Mississippi Gulf Coast User's Group, just to let you know "Amiga Fever" is alive and well in the deep South. We presently have approximately 20 regular members and are gaining strength every meeting. Though most of us are relatively new Amiga users, we are an enthusiastic group and have learned a lot from each other since our beginning in February of this year.

Our unofficial sponsor is Mr. Doyle Brant, President and owner of National Computer Center of Ocean Springs, Mississippi. Doyle is one of your Amazing Dealers and has single-handedly pulled the group together toward the common goal of getting the most out of this amazing computer. I would like to thank Doyle publicly, on behalf of the group, for being a truly "Amazing Dealer." We really appreciate all of his efforts.

We would also like to thank you, Amazing Computing, for producing such an informative magazine. Your publication has time and again helped many of our members understand more about this fantastic digital powerhouse called the Amiga.

Sincerely,
Allen R. White
Vancleave, MS

P.S. We certainly welcome all interested parties to our meetings. For more information call Doyle Brant at National Computer Center in Ocean Springs at 875-3142.

As we have often said, a good user group is the best peripheral your computer could have.

Dear Sirs;
I am very much in the hopes that Amazing Computing could help me, as well as many other Amiga users, with the matter of a "Custom Print Driver." Is there anyone or any computer out there, that can write a truly universal Print Driver, with the following features:

1. A Patch Menu to be able to set "Escape features" to match any printer. User selectable.
2. A user setable timer/buffer (espe cially important for my application), so that the system could dump 2K worth of data to the printer at a time. That way a "dumb printer" wouldn't overflow its buffer.
3. Be compatable with V 1.2 software and standard Word Processing software.
4. Incorporate a "rearranging text" subroutine to allow an 80-column screen to print up to a 132-column page.

If you can be of any help in this matter, please advise. Your magazine is great; please keep up the great work!!

Sincerely Yours,
Robert J. Sampson
Wetumpka, AL 36092

We are in luck. Please see "All About Printer Drivers" by Richard Bielak on page 67. This will not give all the answers, but it is a great place to start.

•AC•

Software designed for AMIGA®.

Lattice® C Compiler $225.00

New version 3.1 of the AMIGA DOS C Compiler replaces version 3.03. Major enhancements include the addition of: TMU, an assembler, a faster linker and version 3 MS-DOS.

With more than 30,000 users worldwide, Lattice C Compilers set the industry standard for MS-DOS software development. Lattice C gives you all you need for development of programs on the AMIGA. Lattice C is a full implementation of Kernighan and Ritchie with the ANSI C extensions and many additional features.

Professional Lattice® C Compiler $375.00

A new product called the Professional Lattice C Compiler is now available. It includes the C Compiler package (complete with TMU), plus LMK, LSE and the Metascope Debugger.

AMIGA® C Cross Compiler $500.00

Allows AMIGA development on your MS-DOS system. Price includes the Professional Lattice C Compiler described above.

Lattice Screen Editor (LSE™) $100.00

Designed as a programmer's editor, *Lattice Screen Editor (LSE)* is fast, flexible and easy to learn. *LSE's* multi-window environment provides all the editor functions you need including block moves, pattern searches and "cut and paste." In addition, *LSE* offers special features for programmers such as an error tracking mode and three Assembly Language input modes. You can also create macros or customize keystrokes, menus, and prompts to your style and preferences.

Lattice dBC III™ Library $150.00

The *dBC III library* lets you create, access and update files that are compatible with Ashton-Tate's dBASE system. *dBC III's* C functions let you extend existing dBASE applications or allow your users to process their data using *dBC III* or dBASE III.

Lattice Text Utilities (TMU™) $75.00

Lattice Text Utilities consists of eight software tools to help you manage your text files. GREP searches files for the specified pattern. DIFF compares two files and lists their differences. EXTRACT creates a list of file names to be extracted from the current directory. BUILD creates batch files from a previously generated file name list. WC displays the number of characters and optionally the checksum of a specified file. ED is a line editor which can utilize output from other *TMU* software in an automated batch mode. SPLAT searches files for a specified character string and replaces every occurrence with a specified string. And FILES lists, copies, erases or removes files or entire directory structures which meet the specified conditions.

Lattice Unicalc® Spreadsheet $79.95

Unicalc is a simple-to-operate program that turns your AMIGA computer into an electronic spreadsheet. Using *Unicalc* you can easily create sales reports, expense accounts, balance sheets, or any other reports you had to do manually.

Unicalc offers the versatility you've come to expect from business software, plus the speed and processing power of the AMIGA.

• 8192 row by 256 column processing area • Comprehensive context-sensitive help screens • Cells can contain numeric, algebraic formulas and titles • Foreign language customization for all prompts and messages • Complete library of algebraic and conditional functions • Dual window capabilities • Floating point and scientific notation available • Complete load, save and print capabilities • Unique customization capability for your every application • Full compatibility with other leading spreadsheets • Full menu and mouse support.

Lattice MacLibrary™ $100.00

The *Lattice MacLibrary™* is a collection of more than sixty C functions which allow you to quickly and efficiently take advantage of the powerful capabilities of the AMIGA.

Even if your knowledge of the AMIGA is limited, *MacLibrary* can ease your job of implementing screens, windows and gadgets by utilizing the functions, examples and sample programs included with the package.

Other *MacLibrary* routines are functionally compatible with the most widely used Apple® Macintosh™ Quickdraw Routines™, Standard File Package and Toolbox Utility Routines enabling you to rapidly convert your Macintosh programs to run on the AMIGA.

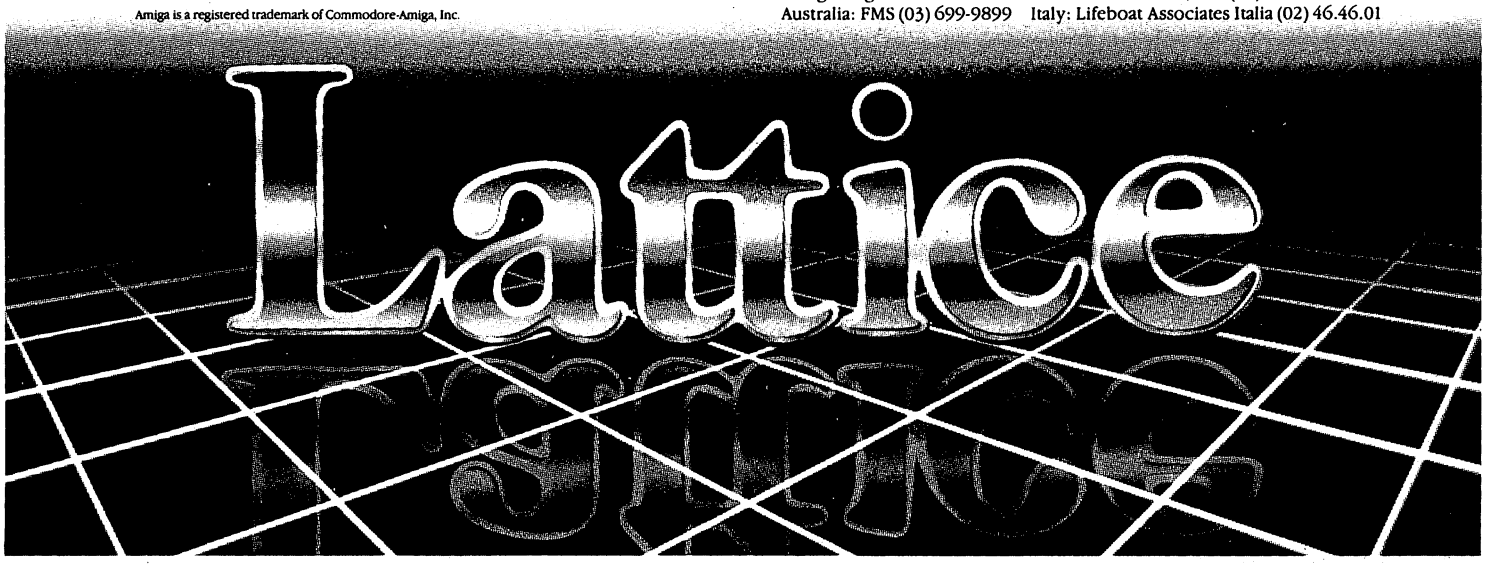
Panel™ $195.00

Panel will help you write your screen programs and layer your screen designs with up to ten overlapping images. *Panel's* screen layouts can be assigned to individual windows and may be dynamically loaded from files or compiled into a program. *Panel* will output C source for including in your applications. A monitor and keyboard utility is also included to allow you to customize your applications for other systems.

With Lattice products you get *Lattice Service* including telephone support, notice of new products and enhancements and a 30-day money-back guarantee. Corporate license agreements available.

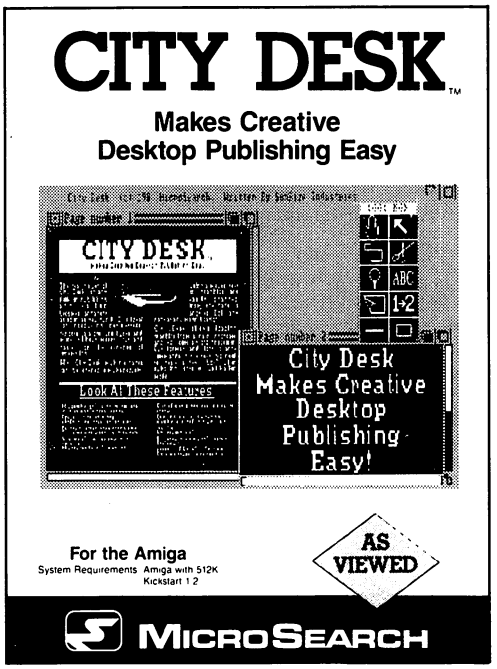
Lattice, Incorporated
Post Office Box 3072
Glen Ellyn, Illinois 60138
(800)533-3577 In Illinois (312) 858-7950
TELEX 532253 FAX (312) 858-8473

INTERNATIONAL SALES OFFICES: Benelux: Ines Datacom (32)2-720-51-61
Japan: Lifeboat, Inc. (03)293-4711 England: Roundhill (0672)54675
France: Echosoft (1)4824.54.04 Germany: Pfotenhaur (49)7841/5058
Hong Kong: Prima 85258442525 A.I. Soft Korea, Inc. (02)7836372
Australia: FMS (03) 699-9899 Italy: Lifeboat Associates Italia (02) 46.46.01



Superior Products

"MicroSearch has a hands-on philosophy for product development. As a retail store and distributor, we constantly research the market. Our formula for success is simple. At MicroSearch we listen to our customers... carefully."



City Desk™ makes creative Desktop Publishing easy. The real power of City Desk for any desktop publishing application becomes apparent as soon as you run it. It is ideal for producing newsletters, reports, catalogs, brochures, and more, without messy "cut and paste" or the expense of typesetting.

With City Desk, multiple pages can be opened simultaneously. Using a mouse, text or graphics are easily dragged from one page to another. Edit any page at any magnification.

City Desk allows loading graphics from a paint program and text from a word processor. Put format and font change commands in the text. No need to create a new "box" when switching fonts or justification mode.

City Desk Features Include:

- Mix graphics and text on the same page
- Automatic and manual kerning
- Powerful escape commands
- Shadows, lines, boxes, and patterns
- Put any number of columns on a page
- Resize and crop graphics with a mouse
- View and edit multiple pages at the same time
- Mix any number of different fonts
- Embedded command codes to allow font changes
- Print color pictures in gray scales
- Graphic editor and library of clip art included
- Not copy protected
- Utilizes extra memory and supports hard drives
- Supports HP LaserJet+, Postscript Laser and any preferences printer

EYE-RESolution "Be kind to your eyes"

The Problem: Annoying flicker in Hi-Res Interlace mode.

The Solution: EYE-RESolution Screen. For only $16.95!

The new "hook and hold" feature allows for a clean, simple attachment. After all who needs messy glue residue or weak velcro?

The real reason for buying EYE-RESolution? Simple... it works!

Perfect Sound is the lowest priced Digital Sound Sampler on the market today. **And it records in stereo.** Both channels... at the same time!

Perfect Sound is **Fun**. You can record sounds, play them back faster or slower, even backwards!

Perfect Sound is **Affordable**.
It's only $89.95!

Perfect Sound SPECS.

Two channel digitizer.
IFF compatible.
Instrument editing.
"C" Source code included.
Software not copy protected.
Graphs, file compression, and more!

Amazing Computing, Volume 2, No. 5, Said, "I highly recommend the PERFECT SOUND digitizer."

MICROSEARCH

9896 Southwest Freeway • Houston, Texas 77074
713-988-2818, 9am-6pm CST.

See your Amiga Dealer today, or call us for the dealer nearest you. Dealer inquiries welcome.

The New Breed of Amiga *Video* Products

by John Foust

Amiga Video

A recent issue of Business Week magazine forecast a booming segment of the computer market: desktop video. The Amiga was featured at center stage, up against systems that cost four to five times as much.

This summer marks the appearance of many new video products for the Amiga. These include Sculpt 3-D from Byte-by-Byte, VideoScape 3D, Aegis VideoTitler from Aegis Development and new video hardware from Mimetics.

Sculpt 3-D

Sculpt 3-D creates three-dimensional objects and then renders an image of that object, viewed in perspective, from any angle or lighting. The image can be made in HAM 4096 color mode, so the images are very realistic. Imagine Aegis Draw extended to three dimensions, cross it with the Juggler demo and you'll have an idea of Sculpt 3-D.

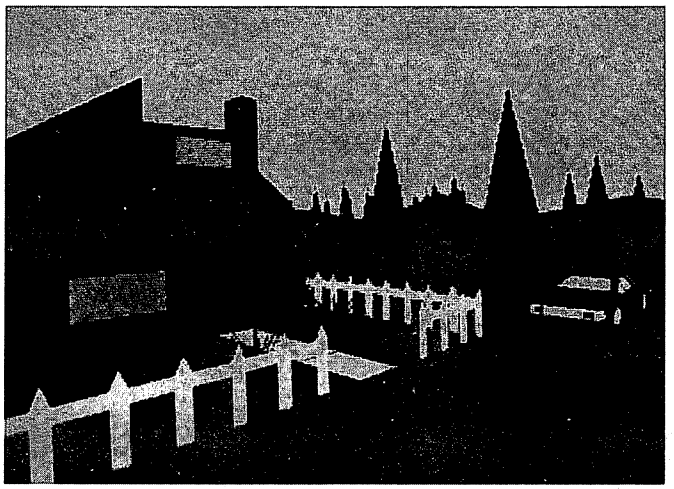
Sculpt 3-D was created by Dr. Eric Graham, who also made the Juggler demo. His Amiga ray tracing software was featured in a recent issue of *AmigaWorld*.

All Sculpt 3-D objects are created from triangles. The object editor has a number of pre-built shapes that can be loaded and resized (such as prisms, cones, spheres and hemi-spheres, all composed of joined triangles.) Shapes can be joined to create complex objects. A magnet tool can pull and distort the selected faces of objects. The triangles can be colored independently and assigned a texture, such as dull, shiny or mirror.

Graham is a photographer, so the program has a camera-like interface. You select the observer's position and point the camera at a spot. Lights can be positioned to illuminate the object.

The program multitasks, so you can continue to create objects in the editor while the scene is being rendered. The ray tracing technique used to create the life-like images takes several minutes to several hours to create an IFF image, depending on the complexity of the objects in the scene.

Byte-by-Byte hopes to market an animation package later this summer. Simple animations can be made with the basic Sculpt 3-D package, but a more complex system is being developed.



VideoScape 3D

VideoScape 3D is a three-dimensional animation program. It has three parts: an object editor, an animation sequence creator. and the animator program itself. Objects are created out of polygons in the object editor, then scenes of objects are composed in the animation sequence editor. The camera viewpoint is recorded in this process, as well.

Finally, the whole sequence is played back with the animation program. Animations can be played back in simple wire-frame renderings for speed or in full color, which takes a little longer. The animation frames are rendered in any resolution except HAM. With dithering, the program has effectively put about 50 colors to use when making frames.

VideoScape 3D was created by Allen Hastings. He is a graphics programmer at the Lockheed Missiles and Space Company in Sunnyvale, California. The program has a lot of flexibility, according to Hastings: "If you don't want to do animation, you can do perfect perspective drawings."

Hastings is working with Aegis to produce a new IFF format for animations called ANIM. With an ANIM recorder program, almost any program can be used to produce animations. Single frames are recorded and only the difference between consecutive frames is stored in the IFF file to reduce storage requirements.

continued...

Aegis VideoTitler

VideoTitler is being produced by Sparta Software. It creates static title screens for presentation slides, television or video production. It has methods for manipulating fonts not found in any other Amiga titling program.

It is a two program set. The first program creates the static images; the second program coordinates a "slide show" of a series of static images. This program can work in cooperation with ANIM animation sequences created with VideoScape 3D. It is expected to be released sometime near the end of the summer.

According to programmer Gary Bonham, there is a lot of power in the way VideoTitler manipulates fonts. In addition to standard Amiga fonts, the program supports the new color font standard, as well as a new type of font called vector or polygon fonts. Vector fonts are much more memory efficient than standard Amiga fonts. Large font sizes in standard Amiga fonts use a lot of memory, while vector fonts use much less.

VideoTitler supports all video resolutions, as well as overscan and the extra half-bright mode of some Amiga graphics chips. Complex backgrounds can be created by tiling an image or text across the backdrop. Brushes can be twisted and redrawn in many ways, similar to the perspective option in Deluxe Paint II.

Vector fonts can be resized and replotted in many ways. VideoTitler should have six or seven vector fonts in the production version. With permutations available with the styling options in the program, these fonts should go a long way.

VideoTitler can save the arrangement of a screen as an IFF file. The file includes position information for the text and fonts in it. Later, the screen can be reloaded. Because of a new IFF format, the text can be re-arranged without affecting the background.

The second program is called VideoSEG, short for "video special effects generator." This choreographs a series of static frames, changing between frames with a variety of wipes. It can work with the ANIM animation sequences produced by VideoScape 3D, so that static title frames can fade to the start of an animation sequence.

Bonham reports Sparta has produced continous animation lasting six to seven minutes, which was recorded direct to video without stopping the recorder. These sequences included static titles between sequences of animation. They were produced on an Amiga with 5 megabytes of memory and a hard disk.

Mimetics

Mimetics, more well-known for the Soundscape music system, is branching into video peripherals as well. At Spring COMDEX, they showed a low-cost genlock called ImageGen. It works with the Amiga 500 as well as the 1000 and 2000. It generates color composite for the Amiga 500, so it should be popular for that purpose alone. Imagen does very basic genlocking, with no audio mixing. It does not re-route the mixed output back to the monitor of the computer; the genlocked output is only supplied as composite-out. For most applications, this is just fine, according to Bob Hoover at Mimetics. They hope to sell it for about $150. They are also working on a higher-end genlock for professional use.

The most exciting product from Mimetics is a frame buffer. This is an alternate graphics output device for the Amiga. It accepts image data in memory from the Amiga and displays it with professional video quality. The Mimetics frame buffer has 640 by 480 pixel resolution with 24 bits of color (8 bits each for red, green and blue) for a total of 16 million colors on-screen at once. The buffer holds a single static image, and outputs NTSC RS-170A broadcast video.

The software can accept Digi-View DGVW RGB files. Digi-View actually records seven bits of information, so with this frame buffer, you can display more than 4096 colors.

The frame buffer will be available as both an Amiga 2000 Zorro board and an IBM PC board, hopefully for about $700. Mimetics is considering a $200 digitizer daughter board that would capture video images as well as display them.

According to Hoover, this board "gets rid of the barrier between the Amiga and full TV production. Professionals are going to have a hard time arguing about this. It makes a great color bar generator."

Other devices planned by Mimetics are a SMTPE interface and a step-frame controller for Sony 1000 and 3/4 inch video tape decks.

...and MUCH, MUCH MORE!!!

Electronic Arts will soon announce a new Deluxe Paint II Art Disk. It will have a new slideshow program with many, many new features. It will be able to handle full video overscan images, as well as pictures larger than the display size. In this case, the slideshow program can be programmed to pan across the image, sliding the picture over the display from corner to corner or side to side. It should have interactive abilities as well.

There have been several sightings of an improved version of Deluxe Paint II, but no official recognition of this program from Electronic Arts. The update fixes a few minor bugs and adds support for a hard disk named "DH0:", as opposed to the "DH:" in the original Deluxe Paint II. One bug supposedly caused hard disk crashes when a subdirectory of pictures had more than 40 files.

Rumors have been circulating of a consumer version of Caligari, priced around $300. This video animation system produced videos similar to those from Allen Hastings. The company had some financial troubles. They originally intended to market a more expensive animation program, but are now considering a lower-priced consumer version.

•AC•

Very Vivid!

"Very Vivid! began as a company aimed at providing new creative tools for performers, dancers and musicians. This goal remains Very Vivid!'s primary focus."

by Tim Grantham

Six years ago, John Vincent, a performance artist with a psychology degree from the University of Waterloo approached Frank MacDougall, a student in the school's world-renowned computer science program, for technical assistance. Vincent wanted to integrate high technology with choreographed movement. His initial idea was to trigger various devices - VCRs, projectors, synthesizers, etc. - by interrupting laser beams with his body. MacDougall listened, reviewed the literature, thought about it and announced that a computer-controlled video interface would offer far more flexibility and ease of use.

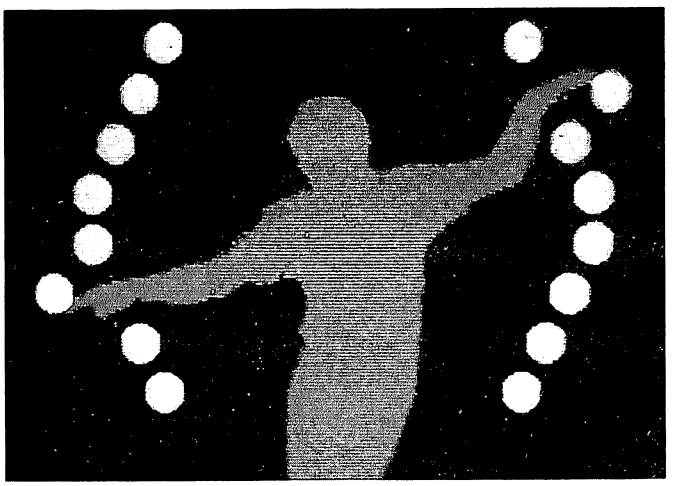
The specialized hardware required for such a system was prohibitively expensive, though - certainly not within the reach of struggling artists. Artificial Reality Corporation, for example, provides a VAX-based system with similar capability for a mere $250,000.

The Amiga and its custom graphic coprocessors, however, has suddenly made the dream reality. The Amiga provides a number of important features: bit-mapped graphics, a color lookup table, direct memory access, a substantial amount of memory and a video sync line to combine the output of the Amiga and another video source.

Vincent and MacDougall teamed up with David Bray, another Waterloo student, completing a graduate degree in economics. His thesis had been on the marketing of innovative products. What MacDougall and Vincent envisioned fell into that category. Bray was intrigued and started extensive research on the music and video markets. At the same time, MacDougall began developing ideas for the software. Bray is now Director of Marketing for Very Vivid!, the company that grew from this unlikely collaboration.

The Live! video digitizer, from A-Squared, provided the crucial interface between the camera and the Amiga. Working with a prototype of the unit, MacDougall began writing the software that would not only control the interaction between the digitized image and the Amiga's graphics system, but also provide a support environment for the creation of the animations themselves. The result is the Mandala system.

MacDougall calls the Mandala a 'video sequencer' and the comparison to a music sequencer is appropriate. A musician can trigger a sequence of notes by striking a single key on his synthesizer; the performer using the Mandala can trigger



animations. Multiple drawings, created with a paint program like Deluxe Paint or Aegis Images, are chained together into short sequences. These drawings, in turn, can be combined to form a super-sequence. The order of the sequences during performance is controlled by the Mandala program, a standard ASCII 'script' created by the videographer and the person in front of the camera.

The interaction can go far beyond simple triggering. The Mandala software enables the animated objects to have character, as well as movement. The 'stickiness' of an object can be controlled— when it collides with the digitized image, the designer can cause it to instantly carom away like a billiard ball, or stay attached until a particularly sharp movement of the 'digitizee' throws it off.

The Amiga's multitasking capabilities broaden the possibilities even further. Each event can spawn other events. When a 'hit' of an object is detected, the Mandala software can signal other devices— a synthesizer over a MIDI bus or a laserdisk video player, for example, via SMPTE signals. Even other computers can be controlled. In fact, it goes both ways— all these devices can act as input to the Mandala software. A MIDI keyboard can be used to trigger the animation sequences, for example. Are you starting to grasp the scope of what we're talking about here?

The people at Very Vivid! certainly have taken hold of the situation. New artists, both of software and graphics, are being hired. The company has just moved into a spacious

continued...

studio. Discussions are ongoing with a number of other companies, manufacturers of hardware mostly, exploring the possibilities of joint ventures.

Very Vivid! began as a company aimed at providing new creative tools for performers, dancers and musicians. This goal remains Very Vivid!'s primary focus. For $30,000, the artist purchases a turnkey system that, in one package, provides everything he needs— multiple MIDI interfaces, hard disk, SMPTE interfaces, the digitizer, lots of memory, a 68020/68881 board and a Genlock. Included in the cost of the system is a week's training, the services of Vivid!'s graphic artists and extensive technical support.

This product is obviously intended for major artists with the skill for this kind of multidisciplinary tool (like Laurie Anderson or Talking Heads). Operators of theme parks (like Disneyland), perhaps, could put the Mandala to use in high-tech recreational environments. Don't worry, though! Very Vivid! does have plans for less sophisticated products; ones you and I can afford.

Very Vivid! decided, after interminable delays in production of the Live! digitizer, to design and manufacture their own digitizer. This unit also plugs into the Amiga's expansion port. It will be made available on a Zorro card for the Amiga 2000.

For about $1500, you will be able to buy the digitizer and the Mandala 1.0 software. Mandala 1.0 is an icon-based version of the 'Super Mandala' software that comes with the turnkey system. While it is not a comprehensive package, Mandala is a professional package that provides a simpler, mouse-driven interface and permits your digitized image to control MIDI synthesizers, change screens on the Amiga and interact with simple animations. Like Mimetics' Pro MIDI Studio software, Mandala's capabilities can be expanded by adding modules written by Very Vivid! and others. The hefty price quoted is only for those who can't wait: the digitizers are now being hand-assembled. Once mass-production begins, prices will drop.

Much software for the digitizer is currently under development, including a program called BodyPaint. As you might guess, this paint program lets you use your body as a video brush. Another program in the development stage provides images of musical instruments you (or rather your digitized image) can play. These images trigger sounds created by the Amiga's own sound chip. Yet another program aims to provide a new approach to the ancient video game Pong! You can literally kick a ball around the screen. The ball has character, if not intelligence. It can be sticky or slick, heavy or light, big or small. The ball can be affected by gravity (which *you* can control). You can also have more than one ball on the screen. All these programs will sell at competitive prices.

The phrase 'interactive adventure game' will take on a whole new meaning. A demo of one game showed a player literally creeping down the halls of a dark castle, falling down a pit, running through murky catacombs and being trapped in a dungeon as a grate slammed down behind him! All these castle images have been created by Very Vivid!'s artists. The player made the appropriate motions in front of the camera. Add iconic images as part of the scene and the player can directly control the graphic action of the game. In the castle, for example, the player opened a door by pushing a brick in the wall.

There's no reason why the Mandala software, combined with the digitizer, a Genlock unit and the appropriate controller interface, cannot interact with video imagery from a laserdisk (as is done in a more crude fashion, in certain arcade games). In less than two years, you may be able to pick up your favorite group's latest compact disk, containing both video and sound, bring it home to your Amiga and actually become a part of their music videos!

The possibilities are endless. Imagine a security system that can not only detect when someone enters a room, but can tell when someone is in a particular area of a room! If you want to change the secured area, just color over its image on the Amiga's screen. Imagine a program that lets a quadraplegic get back some control of his environment simply by moving a finger or blinking an eye. Or athletic training programs: check your karate style as you let loose a flying kick at an evil icon. I leave the rest to your imagination.

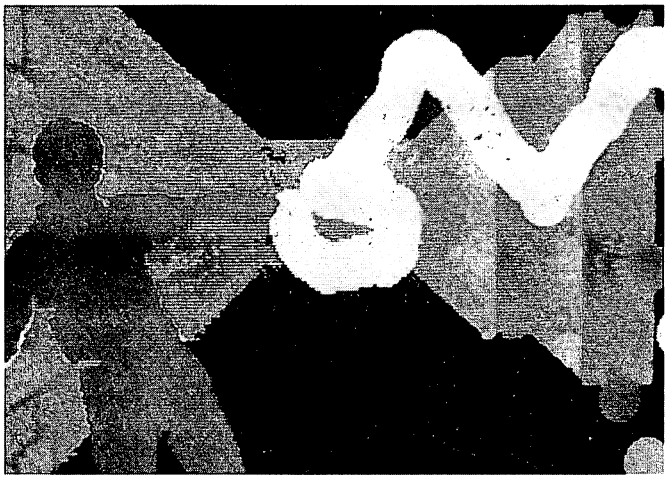
The people at Very Vivid! are understandably reluctant to go into great depth about how their products work, but the principles are fairly straight-forward.

The Mandala software uses a 320 by 200 pixel, 32-color display. This layout requires five bit-planes in the Amiga's graphics memory. The video image is placed in the bottom plane by the digitizer. The other graphics are generated by the Mandala program run on the upper four planes. Because of this, these graphics are limited to sixteen different colors. The Mandala software can detect when a four-plane graphic object encounters the single bits of the digitized image in the fifth plane [see drawing]— the color for that pixel is then set by a different color register.

Upon request from the software, the board places one frame of the video signal into CHIP memory. The Mandala software then updates the software-generated part of the display, as laid out by the animation designer. The more complex the animations and interactions are, the longer this process takes. This process can cause the movement of the digitized image to become jerkier, as the period between image requests becomes longer. MacDougall tells me that upgrading to a 68020 CPU and 68881 math coprocessor significantly speeds up the processing of the animations.

The digitizer is monochromatic only. In fact, having 16-color graphics forces the digitized image into what MacDougall calls 'shadow mode' because it can take up only one bit-plane (2 colors). Presumably, more color can be shown in the digitized image, but only at a cost of color in the animated graphics.

There's an easier and more dramatic way of getting around this limitation, though. Simply feed the video signal into the digitizer AND the Genlock unit. The digitized image (flipped back to normal orientation) lies exactly over the video image



on the monitor. The software then makes the digitized image invisible. The Genlock signal replaces the background color on the monitor. The subject is then in full color, interacting with 16-color graphics.

The digitizer works best in a high contrast situation. Vincent, during his performances with the Mandala, wears an off-white body suit and stands before a plain, dark background.

Very Vivid! has been careful to play by the rules. IFF standards are followed. The Mandala software does not prevent other tasks from running. The digitizer auto-configures. Very Vivid! also makes the library of custom routines that Mandala uses available to outside developers (In this respect, Very Vivid! is following in the footsteps of Mimetics' Soundscape software). This approach is yet another example of the highly integrated, modular approach of the Amiga and its software. . .which should be applauded and encouraged.

The Mandala's live debut took place at the World of Commodore IV in Toronto—a stunning demonstration designed and performed by John Vincent. The live musical accompaniment was also written and performed by the Very Vivid! staff.

Other companies, including A-Squared, will also be producing digitizers. Bray says that the low-priced line of games and utilities Very Vivid! intends to produce will be compatible with these digitizers. These digitizers may or may not be superior to the one made by Very Vivid, but it's important to note that Very Vivid! has a jump on other developers in terms of software. Very Vivid! has been preparing and developing for years. Very Vivid! provides the unique combination of multidisciplinary skills required to fully exploit the possibilities of a digitizer.

•AC•

Very Vivid!
1499 Queen St. W., Suite #302
Toronto, Ontario
Canada M6R 1A3
(416) 537-7222.

ATTENTION! DELUXE MUSIC CONSTRUCTION SET™ USERS

ATTENTION! MUSIC STUDIO™ USERS

SYMPHONY SONGS

ATTENTION! INSTANT MUSIC™ USERS

SYMPHONY SONGS gives you a library of nearly 1,000 music masterpieces. All songs are in IFF format so they may be loaded, played, printed, transposed, and modified in any way you like using your favorite composition program. Included is a free program to convert the IFF files to MUSIC STUDIO™ format.

The songs have been arranged by C. Clark Rulaford and Randy Spector and take advantage of the full **4 Voice** capability of the AMIGA.

Space does not allow listing all the songs in each of the volumes. We have listed a few and show the total number in each volume as well as the playing time. A complete list of songs may be purchased for **$3.95.** Each volume of the 27 volumes listed is **$24.95.**

BEATLES Part 1
Vol 15 (21 Pieces 40 Min)
Let It Be, Yesterday, Eleanor Rigby, When I'm 64, . . .

BEATLES Part 2
Vol 40 (17 Pieces 40 Min)
Magical Mystery Tour, Lucy In The Sky With Diamonds, Penny Lane, . . .

CLASSICAL Part 1
Vol 27 (19 Pieces 40 Min)
Prelude #1, Moonlight Sonata 1st and 2nd Movement, . . .

CLASSICAL Part 2
Vol 34 (15 Pieces 40 Min)
Sonata In C Major, Jesus Joy Of Man's Desire, . . .

CLASSICAL Part 3
Vol 31 (18 Pieces 35 Min)
1st Piano Concerto, Polonaise Sonata In C Major, Etude #3, . . .

CLASSICAL Part 4 (Bach)
Vol 35 (22 pieces 30 Min)
Two Part Invention #1, Three Part Invention #6, Prelude and Fugue 1, . . .

CLASSICAL Part 5 (Bach/Clementi)
Vol 46 (24 Pieces 50 Min)
Choral #1, Sonata #1, Theme and 11 Variations From The 2nd Sonata, . . .

BEETHOVEN, BROADWAY, & BLUES
Vol 38 (15 Pieces 40 Min)
2nd Movement Of the Pathetique Sonata, Minuet In G, Fuer Elise, . . .

COUNTRY CLASSICS Part 1
Vol 41 (18 Pieces 45 Min)
Thank God I'm a Country Boy, Act Naturally, . . .

ROCK Part 1
Vol 32 (19 Pieces 50 Min)
AXEL F, Eye Of The Tiger, Both Sides Now, . . .

ROCK Part 2
Vol 16 (21 Pieces 40 Min)
Georgy Girl, Guantanamera, Theme From "Love Story," Cherish, . . .

80's GREATEST
Vol 24 (16 Pieces 50 Min)
Hill Street Blues Theme, Chariots Of Fire Theme, Dynasty Theme, . . .

70's GREATEST
Vol 12 (21 Pieces 45 Min)
Tie A Yellow Ribbon On The Old Oak Tree, We've Only Just Begun, . . .

60's GREATEST
Vol 13 (21 Pieces 45 Min)
Windy, By The Time I Get To Phoenix, Come Saturday Morning, . . .

GOLD & PLATINUM HITS
Vol 45 (19 Pieces 60 Min)
Thriller, 99 Luft Balloons, California Girls, . . .

KENNY RODGERS HITS
Vol 39 (12 Pieces 45 Min)
Lady, Ruby, She Believes In Me, The Gambler, . . .

BILLY JOEL GREATEST HITS
Vol 43 (17 Pieces 65 Min)
Piano Man, Say Goodbye To Hollywood, Only The Good Die Young, . . .

COUNTRY CLASSICS Part 2
Vol 42 (19 Pieces 50 Min)
Ode To Billy Joe, Me and Bobby McGee, Country Roads, . . .

TV THEMES
Vol 37 (21 Pieces 35 Min)
Hill Street Blues, St. Elsewhere Theme, Masterpiece Theater Theme, . . .

MOVIE THEMES
Vol 19 (23 Pieces 40 Min)
MASH Theme, The Rose, Can You Read My Mind (Superman), . . .

BROADWAY'S THEMES
Vol 47 (25 Pieces 65 Min)
The Last Supper, Dr. Doolittle, The Old Dope Peddler, . . .

CHURCH MUSIC
Vol 28 (26 Piece 50 Min)
Amazing Grace, What A Friend We Have In Jesus, . . .

BARBERSHOP
Vol 22 (22 Pieces 45 Min)
Hello Dolly, Put On a Happy Face, Hey Look Me Over, . . .

RICHARD RODGERS SONGBOOK
Vol 18 (19 Pieces 40 Min)
Climb Every Mountain, DO-RE-MI, The Sound Of Music, . . .

NOSTALGIA
Vol 17 (22 Pieces 45 Min)
Let Me Call You Sweetheart, Ain't Misbehavin', On The Goodship Lollipop, . . .

CHRISTMAS
Vol 36 (26 pieces 50 Min)
O Little Town Of Bethlehem, Let It Snow, March Of The Toys, . . .

POLKA PARTY
Vol 33 (18 Pieces 40 Min)
Happy Polka, Pizzacato Polka, Betty Polka, . . .

SYMPHONY JUKEBOX

Symphony Jukebox allows you to program a selection of songs, their order as well as the number of times, and allows you to listen to them for hours of uninterrupted playing. Other features include MIDI output, instrument selection, transposition, and tempo modification. **$24.95**

SYMPHONY MUSIC VIDEO

This program has all the features of our SYMPHONY JUKEBOX, however, it also allows you to specify a picture to be displayed with each song. The pictures and music are all in standard IFF format so you may use the songs and pictures included, or use those you developed with your music program (i.e. DMCS), or your paint program (i.e. Deluxe Paint). Included are Christmas music and pictures. **$24.95**

We accept CASH, CHECK, COD, VISA and MASTER CARD orders.
Shipping and handling US and Canada $3.00
Shipping and handling outside the US and Canada $5.00
COD charge $2.00
Illinois residents add 6¼% sales tax.

VISA MasterCard
Dealer Inquires Invited

Speech Systems
38W255 DEERPATH ROAD
BATAVIA, ILLINOIS 60510
(312) 879-6811

Deluxe Music Construction Set, Deluxe Paint, Instant Music are trademarks of Electronic Arts. Music Studio is a trademark of Activision.

VIDEO AND YOUR AMIGA

by Oran Sands, III

The Amiga is a wonderful graphics machine. . . but what use are the graphics, other than in games? Desktop Video! The latest in computer catch phrases. The term implies the ability to create entire videos at your desk. This capability is just a small part of the Amiga's video uses.

The Amiga is not only well-suited to video applications—— it was designed with professional video in mind. There are several clues in both the hardware and software design of the machine to its clear video slant. In the first part of this article, let's discuss the video capabilities of the Amiga. Later, we'll talk about video applications for the Amiga.

NTSC RS-170

Both your television and VCR use a composite video signal that adheres to a standard known as NTSC RS-170*. Interestingly enough, the computer industry has avoided this standard. In the beginning, the computer industry was off target due to ignorance; later, trying to implement high resolution graphics (which can be difficult in standard video) caused problems. The end results were computer systems with poor video outputs.

The Amiga, however, is a pleasant change from all that! The RS-170 signal standard specifies that color is encoded using a subcarrier frequency of 3.58 MHz. The Amiga's clock frequency is exactly double this figure. Since all video timing signals can be derived from the 3.58MHz signal, the Amiga has easy access to the precise signals necessary for proper video. The RS-170 standard also specifies that all signals must be interlaced. This mode is known as the 320x400 and 640x400 graphics states on the Amiga. Although affectionately cursed as the "flicker mode," this interlace is all important in providing the correct signal for video production. We'll discuss how to avoid the flicker later.

PIXELS AND SCANLINES

The NTSC video signal is displayed by scanning one full picture every 30th of a second! The scanlines are traced across the screen, left to right, top to bottom. . . all 525 of them. What? You thought the Amiga had a resolution of 400 lines? Don't confuse "scanlines" with "pixels." They just don't mean the same thing. Resolution, when used with video descriptions, varies according to what it is describing.

The NTSC composite video signal always has 525 scanlines per full picture frame. For instance, a video camera has a rated resolution, depending on how many alternating dark lines it can discern. Computer graphics have a resolution equal to the maximum number of individual pieces (pixels or picture elements) it can generate to provide a full display (i.e. 640 pixels horizontally by 400 pixels vertically).

On the Amiga, this full display isn't a full screen (don't forget that the border is computer-generated also!). Remember, the computer deals with on/off signals, while the television signal may vary infinitely between preset levels. It is this infinite variability that allows for the smooth shading we try so hard to recreate on our computers. The conversion of the computer on/off data to an analog video signal is where the 640x400 pieces are "translated" to a grid of 525 lines. Now for the confusing part.

INTERLACE

In the early days of television, the tv signal scanned 525 lines, top-to-bottom, in a 1/30 of a second. A problem occured when motion was shown on the tube. In the time taken to scan one full picture, (1/30th sec.), the subject had already moved and the next picture started with the subject farther ahead than expected. A "smear" or "jump" resulted.

Interlace was developed to solve this problem. A faster scan rate would have been the right answer, but the technology of the day wasn't quite up to it (not to mention what to do to ensure compatibility with all existing sets). So, developers decided to scan all the odd-numbered scanlines first (in half the time, 1/60th of a sec.). The total scantime for one full picture was still 1/30th of a second. Everthing was fine. The even-numbered scanlines were offset by one-half a scanline, so that the odd and even scanlines wouldn't overlap each other. Perfect.

So, what's the problem with the Amiga? The Amiga stores one bitmap to describe the screen in non-interlace. This one bitmap is scanned twice in a 1/30 sec., with the scans put directly on top of each other. In interlace modes, two bitmaps are used, one containing the even-numbered lines

continued...

(2,4...398,400), one containing the odd-numbered lines (1,3...397,399). The first bitmap is displayed and, while it's decaying, the second bitmap is shown. Each bitmap is "refreshed" only once every 1/60th sec. These bitmaps are then offset as described above. Non-interlace modes don't offset.

The Genlock 1300 does a cute trick. The Genlock puts out an interlaced picture from a low-res image. How? It uses the same bitmap for both even and odd scanlines. The half-line offset forces the interlace mode, blurring the image, so it looks like a "jitter" of the picture. Hi-res flicker occurs because the first scanlines are dying in intensity, while the new ones are being scanned. The "refresh" rate is 1/30th of a second. When pixel-thin lines remain still, this flicker is exaggerated. Motion tends to smooth things out.

Motionless, thin lines (like the ones in our low res image) just don't occur in normal television. So, how do you solve the jitter problem for computer graphics? Simple, don't create pixel-thin objects! Unless you're doing CAD-CAM work, such lines are not needed. If you must use pixel-thin lines, choose a color that doesn't contrast with the background. You can drastically reduce the flicker in this way. Try different combinations to find one you like.

GENLOCK AND SYNCHRONIZATION

If you plan to use the Amiga in a studio environment, you have two choices: "standalone" or "system" connections. To use the Amiga in a standalone configuration, simply connect the Amiga output directly to the device you wish to use it with (a VCR, for example). Such a connection isn't always possible, as today's special effects equipment requires all video signals to be synchronized. Many video devices allow inputs, so that signals can be applied to ensure "sync." The Amiga also allows for such synchronization. Pins on the DB 23 video connector allow such signals to reach the Amiga.

The process of syncing your computer to another video signal is called genlocking. Does the name ring a bell? Once two video signals are genlocked, you can acheive special effects with your computer and external video device.

The GENLOCK unit (provided by Amiga) not only syncs the Amiga, but also performs a special effect known as "keying". Similar to the "blue screen" process used for movies, keying lets you to drop sections of one picture and replace these sections with corresponding portions of another picture. The selection of which section must be dropped is often made on the basis of luminance (brightness) or chroma (color). The GENLOCK makes its decision on the basis of color, always dropping out color #00 (the usual blue background on a typical Workbench screen). [Note to video professionals: Keying is not the same as chromakeying. Do not expect to draw a weathermap on a blue background and have your weatherman overlaid on the map. The Genlock allows only for the opposite to occur—— the map overlays the newscaster.] This capability to genlock allows the Amiga to feed into an even more sophisticated system.

Another professional video feature of the Amiga is the analog red, green and blue signals output (with accompanying correct sync signals). These basic signals are combined to create the composite video signal that contains all the information needed to create a full-color picture. Many professional cameras output only the analog RGB signals which are often fed to a broadcast quality signal encoder, producing a composite signal.

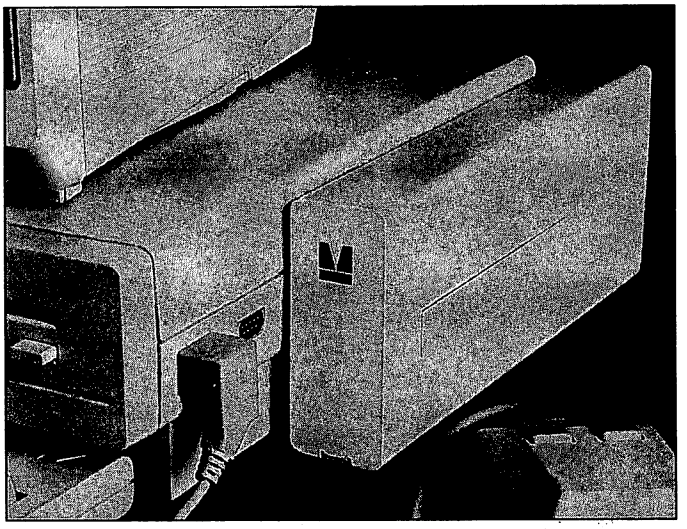
Many high-quality video projectors have inputs for separate RGB analog signals. In fact, our local user group uses such a projector for their meetings in precisely this way. The encoding process must be undone at the monitor or display device. A loss of quality is inherent in both the encoding and decoding process. Since it lets us use the best signal for the application, the ability to provide RGB signals is a real plus for the Amiga!

A FICTICIOUS VIDEO PROJECT

The Amiga has great potential for video presentations and applications, including desktop video. How can the Amiga be used to create desktop videos, you ask? Well, let's watch Bob, a ficticious video producer with an all-too-real video project, as he works the Amiga through a desktop video project.

AVAILABLE NOW!
StarBoard2

Auto-Configuring Fast RAM
Zero Wait States
User Expandable from 512k to 2 Megabytes
Bus Pass-Through
MultiFunction Option: battery/ clock, FPU, parity, Sticky-Disk



If you've owned your Amiga® for a while now, you *know* you definitely need more than 512k of memory. You probably need *at least* double that amount...but you might need as much as an additional two megabytes. We want to urge you to use **StarBoard2** as the solution to your memory expansion problem –and to some of your other Amiga-expansion needs as well!

It's small, but it's BIG–

Since most of you want to expand your Amiga's memory without having to also expand your computer table, we designed **StarBoard2** and its two optional "daughterboards" to fit into a sleek, unobtrusive Amiga-styled case that snugly fastens to your computer with two precision-machined jackscrews.

The sculpted steel case of **StarBoard2** measures only 1.6" wide by 4.3" high by 10.2" long. You can access the inside of the case by removing just two small screws on the bottom and pulling it apart. We make **StarBoard2** easy to get into so that you or your dealer can expand it by installing up to one megabyte of RAM on the standard **StarBoard2** or up to two megabytes by adding in an Upper Deck.

This card has decks!

The basic **StarBoard2** starts out as a one megabyte memory space with 0k, 512k, or one megabyte installed. If you add in an optional **Upper Deck** (which plugs onto the Main Board inside the case) you bring **StarBoard2** up to its full two megabyte potential. You can buy your **StarBoard2** with the Upper Deck (populated or unpopulated) or buy the Upper Deck later as your need for memory grows.

And you can add other functions to **StarBoard2** by plugging in its second optional deck –the Multifunction Module!

StarBoard2: functions five!

If we count Fast Memory as one function, the addition of the **MultiFunction Module** brings the total up to five!

THE CLOCK FUNCTION:

Whenever you boot your Amiga you have to tell it what time it is! Add a MultiFunction Module to your **StarBoard2** and you can hand that tedious task to the battery-backed, real-time clock/calendar. A small piece of MicroBotics software in your WorkBench Startup-Sequence reads the clock and automatically sets the time and date in your Amiga. And the battery *is* included (we designed it to use an inexpensive, standard AAA battery which will last at least two years before needing replacement).

THE FLOATING POINT FUNCTION:

If any one aspect most characterizes the Amiga it's *fast* graphics! Most graphic routines make heavy use of the Amiga Floating Point Library. Replacing this library with the one we give you with your MultiFunction Module and installing a separately purchased Motorola 68881 FPU chip in the socket provided by the Module will speed up these math operations from 5 to 40 times! And if you write your own software, you can directly address this chip for increased speed in integer arithmetic operations in addition to floating point math.

THE PARITY CHECKING FUNCTION:

If you install an additional ninth RAM chip for every eight in your **StarBoard2**, then you can enable *parity checking*. Parity checking will alert you (with a bus-error message) in the event of any data corruption in **StarBoard2**'s memory space. So what good is it to know that your data's messed up if the hardware can't fix it for you? It will warn you against saving that data to disk and possibly destroying your database or your massive spreadsheet. The more memory you have in your system the more likely it is, statistically, that random errors will occur. Parity checking gives you some protection from this threat to your data residing in Fast RAM. Note that the Amiga's "chip" RAM cannot be parity checked.

THE IMMORTAL MEMORY DISK FUNCTION (STICKY-DISK):

When you've got a lot of RAM, you can make nice big RAM-Disks and speed up your Amiga's operations a lot! But there's one bad thing about RAM-Disks: they go away when you re-boot your machine. Sticky-Disk solves that problem for you. It turns all of the memory space inside a single **StarBoard2** into a Memory Disk that will survive a warm-reboot! When your Amiga attempts to grab a **StarBoard2** in Sticky-Disk mode, a hardware signal prevents the system from acquiring the **StarBoard2** as FastRAM (and thereby erasing your files) –instead it is re-recognized as a Memory Disk and its contents are preserved intact. If you want to work rapidly with large files of data that are being constantly updated (such as when developing software) you can appreciate the Sticky-Disk!

Fast RAM –no waiting!

StarBoard2 is a *totally* engineered product. It is a ZERO WAIT-STATE design, auto-configuring under AmigaDOS 1.2 as Fast RAM. Since AmigaDOS 1.1 doesn't support autoconfiguration, we also give you the software to configure memory in 1.1.

Any applications software which "looks" for Fast RAM will "find" **StarBoard2.** And you'll find that your applications run more efficiently due to **StarBoard2** on the bus.

A passing bus? Indeed!

What good is an Expansion Bus if it hits a dead end, as with some memory cards? Not much, we think –that's why we carefully and compatibly passed through the bus so you could attach other devices onto your Amiga (including another **StarBoard2**, of course!).

The sum of the parts...

A really nice feature of the **StarBoard2** system is that you can buy exactly what you need now without closing off your options for future exapansion. You can even buy a 0k **StarBoard2** (with a one megabyte capacity) and populate it with your own RAM (commonly available 256k by 1 by 150ns memory chips). When you add **StarBoard2** to your Amiga you have a powerful hardware combination, superior to any single-user micro on the market. See your Authorized Amiga Dealer today and ask for **StarBoard2**

SUGGESTED RETAIL PRICING:	
StarBoard2, 0k (1 meg space):	$349
StarBoard2, 0k (2 meg space):	$395
StarBoard2, 512k (1 meg space):	$495
StarBoard2, 1 meg (1 meg space)	$595
StarBoard2, 2 megs installed:	$879
StarBoard2, 2 megs & MultiFunction:	$959
Upper Deck, 0k (1 meg space):	$ 99
MultiFunction Module:	$ 99
also available:	
Standard 256k memory card:	$129
MAS-Drive20, 20 meg harddisk:	$1495
MouseTime, mouseport clock:	$ 50

MicroBotics, Inc.
811 Alpha Drive, Suite 335, Richardson, Texas 75081 / (214) 437-5330

Bob works at a small teleproduction studio specializing in industrial video, such as training films and informational tapes for their clients. Bob's latest project is a tape describing proper methods of assembling a transmission for a local tractor firm. Let's follow Bob's progress.

Tuesday... Bob sits down at the Amiga and starts to story-board his project. Rather than sketching out the different scenes on the normal storyboard cards, Bob's does his sketches on the Amiga using Deluxe Paint. Each picture illustrates a crucial scene in the program and includes the text (at the bottom of the screen) to be read by the narrator. Clients get a better feel for what a tape will look like with storyboards. Just reading a typed script can often be mis-leading. Bob's work picks up much speed up when he decides to use the brush feature to save pieces of his pictures that he uses often. By the end of Bob's first day of solid work, the pictures are done and Slideshow is programmed to display them.

Thursday...With the script now approved, Bob gathers the parts used in assembly, so he can digitize their images. Using Digi-View, Bob saves likenesses of the parts, so he can repaint them using Images. Repainting allows Bob to use correct shapes, while changing the colors to help with recognition of the various parts of the program (Bob admits he isn't much of an artist!). Each part's image is saved as a "window" or "brush," allowing Bob to use Animator to later move the parts on screen.

Friday... Bob goes on location to shoot the backgrounds for the animations. He will return next week to shoot the narrator on-location.

Monday... The client calls to mention that part #53638A has been modified to be part #53638B, creating an extra notch just below the first two on the left side (same height, but twice as long). This part will be used by the time the program is needed, but there are no such parts available at this time. Such a change is the kind of thing producers have nightmares about. Bob, however, merely calls up the image and paints in a new notch.

Tuesday... The client stops in to check out the assembly sequence. Bob shows the animation sequence (with part #53638B, of course). The client is impressed and wonders why Bob chose yellow and blue for the gears. Bob thinks these colors look nice. The client thinks they look like his competitor's colors! Redoing the artwork would normally cause a great delay (and extra cost). Bob just reloads the paint program and changes those two colors to chartreuse and mauve. Easy enough and the client loves it!

Thursday.... The main body of the tape is edited. Only the graphics have yet to be included. Using the Genlock, Bob plays a videotape of the production plant through the device, as a background for the animations. The gears float over the pictures and take their place alongside each other in the correct order. The aluminum parts (shown in chartreuse) snuggle up to part #53638B, like they were made for each other!

Friday... Bob's project includes making a picture board of the assembly to be used by the classroom instructor. Bob uses the same pictures as he used for the videotape, but he uses the IMPRINT! program and a Polaroid Palette film recorder to shoot slides of the images. These slides will be used to make the pictures the instructor needs. Later, the client wants to print a brochure using these pictures. With the same program, Bob will make color separations for the printers.

Saturday... Bob's freelance job as a wedding videographer has him shooting at St. Paul's Cathedral. Bob has already used his Amiga at home to generate a title screen for different portions of the tape. This extra touch has brought in more business (Bob wonders if he should branch out into divorces as well. His wife isn't amused).

Sunday... Bob's daughter has learned to use DPaint and is well on her way to creating her first picture of a doggie. Bob videotapes her as she works, knowing he can insert the picture into the tape later. Many years from now, his daughter will appreciate her early creativity.

Monday... Final approval day. The client has viewed the program and loves it. Only one small problem. His name doesn't appear in the credits and he is worried about his professional image at the plant. Bob used Pro Video's CG1 to generate all the titles and credits, so he merely adds the clients name to the list of those who made this program possible and re-edits the end of the show. The client, his name now immortalized in lights, is estatic! Promises of fame, fortune and maybe another project or two shine in the future. Bob's day is made and his boss is happy. If the production had been done elsewhere, it would have cost much more for the graphics or there may have been no graphics at all!. Bob's boss bought the Amiga so Bob could do his scriptwriting and production budgeting. He had no idea it could do all this! (Bob knew!)

With another successful program behind him, Bob sits back to enjoy a moment with Defender of the Crown. Life is good.

I hope this scenario has shown a few of the great applications for the Amiga within the realm of desktop video. At my studio, I've added the ability to generate hi-res color titles, animate graphics, generate digitized video pictures, program point-of-purchase display slideshows, generate repeating screens of information for the in-house cable tv channels, combine computer graphics with video images——all for less than $3000. The character generation program alone would've cost $10,000, if I had bought video specific hard-ware!

Admittedly, the images are not up to the highest level of quality preferred for broadcasting use. A cable station or corporate (industrial, educational, etc.) video department would make out quite well, though, with the extremely inexpensive, but effective Amiga computer. There is a much Amiga software and hardware still coming out. Many of these products work at narrowing the gap between broadcast and industrial graphics. Exciting times are ahead!

•AC•

Amigas and Weather Forecasting

by Brenden Larson

*The Amiga can **really** be used for television weather graphics!*

Weathermen (or meteorologists) spend hours scientifically analyzing the weather. They also spend much time creating aesthetically pleasing and interesting maps. These maps are no longer designed on the old magnetic boards and easyls with magic markers. Since the early 1980's, most weather graphics have been produced with thousands of dollars of computers. Now, with the advent of the Amiga and its high quality graphics, weather broadcasts can now be produced at a relatively low price.

The story behind weather forecasts and weather graphics is not a short one. This article discusses the many ways in which a meteorologist uses a computer for obtaining weather data and putting it together. We'll also take a brief look at how some of the "monster" machines stack up against the Amiga, and finally, a peek ahead at what some developers are doing for the Amiga and weather graphics.

Weather Data Networks

When a television station or forecasting service buys a weather graphics machine, their decision depends not only on pixel resolution and amount of colors the machine has, but also on whether the system is interfaceable with a data base. A meteorological data base, where much data is stored, is just like any other data base or BBS. This data becomes quickly outdated because the weather changes constantly.

The first step is actually collecting the weather data. In various spots all over the world, a cooperative network records hourly surface weather information. Most data recording locations are airports. A typical measurement includes: temperature, dew point, wind speed and direction, percentage of cloud cover and the estimated bottoms of the clouds observed, as well as the type and amount of precipitation, if any.

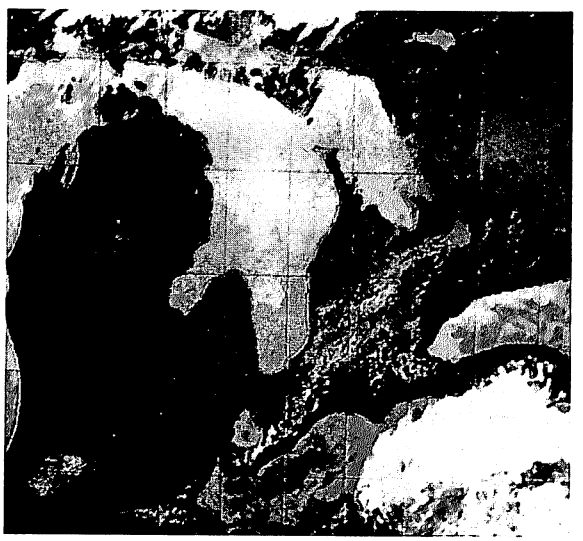
Once these observations are recorded, they are sent to the National Weather Service (NWS) via a special phone or teletype link. Eventually, the data ends up stored in a giant Cray mainframe in Washington D.C., at the National Meteorological Center. Now, with the permission of the NWS, the data can be obtained by private vendors and stored in computers. The data can now be accessed by any person with a computer and the proper software.

Hourly surface observations make up only a fraction of the data available to meteorologists. Also included are parameters measured by weather balloons (which are launched twice a day around the world), numerical forecast models and a ton of specialized info from products produced by the NWS (special aviation reports, heavy snow discussions and technical guidance reports). All this numerical data can be obtained, in ASCII Text format, with a modem and a computer. . . including the Amiga™.

Digital Facsimilie

The super computers at the NWS contour the numerical data into what are commonly referred to as "isobars" and "isotherms" at the top of maps. This transformation gives the meteorologist a more analytical scenario to detect all types of fronts, high pressure zones, jet streams, etc. Once these maps are contoured, the NWS sends out a signal, a digital facsimilie of the original contoured maps, to third party vendors. These signals are intrepreted, stored and dumped to dot matrix printers, so that the meteorologist can make further enhancements for forecasting. For example, Joe Brown at Channel 7 in Chicago eventually receives these facsimilies from one of the vendors through a dedicated computer or printer.

This network is called the Digital Facsimilie (DIFAX) and is one of the most important parts of everyday forecasting. Since the transferred signals are digital, a computer such as the Amiga™ would need proper software and terminal emulation routines to re-display these maps in IFF format. A sample of DIFAX is shown below.

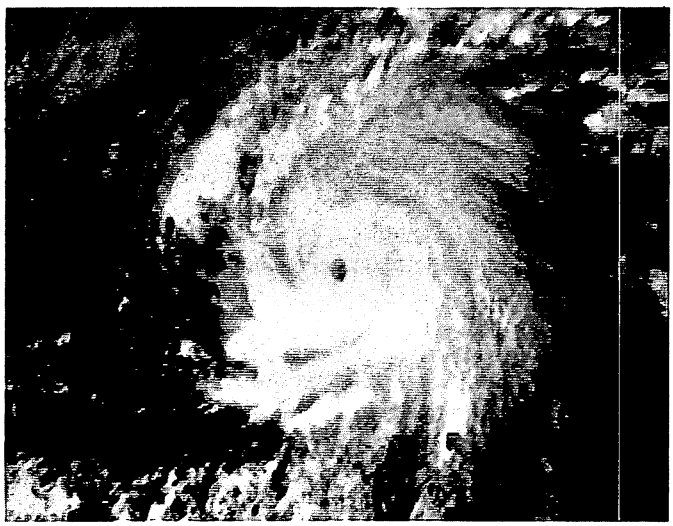


continued...

Goes Satellite

Another important piece of weather forecast data is what all television weathercasts contain...the satellite picture. Taken from a geostationary orbit several thousand miles above the earth, the GOES (Geostationary Operational Environmental Satellite) is one of several satellites that takes snapshots of the earth's cloud, land and sea features. GOES is used frequently for analytical purposes when it comes to forecasting United States weather. Satellite pictures are not only great for on-air television broadcasts, but are also great tools for checking the progression of storms——satellite pictures are received every ten to fifteen minutes! All other data is checked much less frequently and the sooner meteorologists obtain reports, the better the forecast!

One neat application the Amiga accomodates is looping or frame by frame animation of satellite pictures. This feature allows you to see the atmosphere in motion. . . a terrific forecasting tool. Below is an example of a satellite picture produced by a mainframe computer at the NWS and downloaded by an Amiga using special routines. The picture was later converted into an IFF configuration (courtesy of Mr. Dick Garey (NESDIS)).



Weather Graphics Workstation

Using an Amiga as an analytical weather tool is just as important as using an on-air graphics workstation. The Multitasking capabilities of the Amiga allow the system to be continuously updated with weather data, used for weather graphics design, while also being used as a slide projector in a television environment. For the price, the Amiga can't be beat!

Currently only large microcomputers are used in many television stations. These micros are designed specifically as turnkey/workbenches for performing three things:

- Interfacing with a data base
- Desiging weather art
- Putting the art and data together a slideshow format.

So far, these user-specific weather graphics workstations have cost television stations hundreds of thousands of dollars. Smaller budget stations (and there are more smaller budget stations than large ones), can use the Amiga™ to easily perform the duties of a weather workstation.

John Coleman (formerly of Good Morning America and currently at WMAQ-NBC Television in Chicago) uses a machine called an AstroWeather, which produces a global palette of millions of colors (128 at a time), a resolution of 1,024 by 1,024 pixels and interfaces all types of weather data (including satallite pictures, radar maps and DIFAX). Coleman's system automatically dumps DIFAX to a printer. AstroWeather was designed specifically by Weather Services International (WSI), Inc. and costs $90,000.00 to $100,000.00! Of course, a great storage capacity,as well as many other features, enhance AstroWeather.

Tom Silling, meteorologist at cable television's WGN (Chicago), uses a similar computer called the Triton. The Triton produces a global palette of sixteen million colors, a 750 X 500 screen and has many other nifty software features that allow for storage and playback of satellite pictures in loop-animation. Skilling's system can run a whole day's worth of weather (96 frames) in about 2 seconds (around 64 frames/sec)! The cost of the Triton is around $100,000.00 *without* all the extra features! The Triton was developed by Kavouras Company which provides Triton clients with all forms of weather data.

Welcome Amiga

It's true that the Amiga™ has nowhere near sixteen million colors nor a resolution matrix of 1,024 pixels. . . but for a television station with a limited budget, the Amiga can easily become a tool for weather graphics. Two parties are currently working on developing a weather graphics workstation for the Amiga. Personally, I have used the Amiga for some weather work for a cable television affiliate in the Chicago area and the Amiga performed just fine! The only missing ingredient is software to interface the Amiga with third party data bases. Hi-res interlace paint packages have already been developed. They are great tools for producing quality weather art, and a handful of slideshows have been written that tie together television graphics scripts.

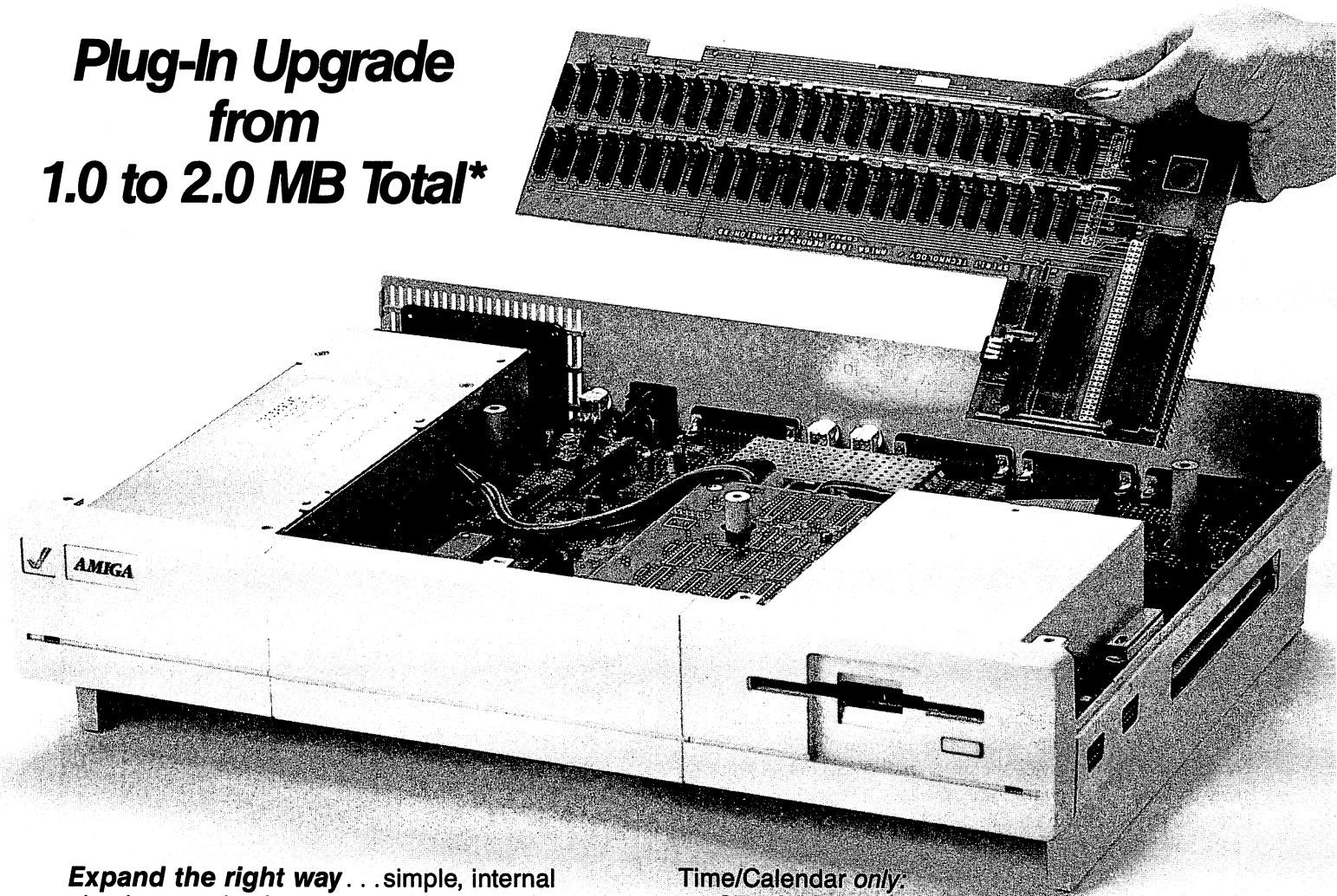
Station Manager

The package most people have already heard about is Station Manager Weather Graphics, SMWG (it has not been officially released by the vendor, Associated Computer Services). I had the opportunity to view a demonstration of an early version of SMWG and it was rather impressive!

Basically, SMWG is a very fancy slideshow program (thus having only one third of what a complete weather graphics workstation should have). It is the first package to allow for double buffered hi-res interlace animation. SMWG also allows for actual sequential animation of raster graphic objects or "brushes." Brushes can represent cold fronts, clouds, suns, snowflakes and dozens of other weather

AMIGA INTERNAL MEMORY EXPANSION Plus TIME/CALENDAR

Plug-In Upgrade from 1.0 to 2.0 MB Total*



Expand the right way. . . simple, internal plug-in mounting leaves your side expansion port free to add other peripherals. Also, the internal Time/Calendar does not use a joystick port.

Memory Expansion Features:

* Zero Wait-State
* No Cuts or Soldering Required
* Full Auto-Configuration
* Lithium Battery Back-Up for Time/Calendar

ORDERING INFORMATION:

DRAM Memory *with* Time/Calendar:

#ST-05	0.5 MB	$349.50 List
#ST-10	1.0 MB	$499.50 List
#ST-15	1.5 MB	$599.50 List

**Memory expansion from 1.0 to 2.0 MB includes AMIGA 1000 512K RAM.*

AMIGA is a trademark of Commodore-Amiga, Inc.

Time/Calendar *only:*

#ST-TC Time/Calendar including Battery Back-Up $59.50 List

ASK ABOUT increased speed with the *new* 68010 Processor

VISA and Mastercard Welcome

CALL TOLL FREE: 1-800-433-7572

Factory direct: 1-801-485-4233

DEALER INQUIRIES INVITED

SPIRIT™ TECHNOLOGY

220 West 2950 South • Salt Lake City, Utah 84115

symbols. Brushes can be brought into a tween (just like Aegis Animator) under a sequence imbedded in a storyboard. The brush can be "told" to move to ten pre-assigned (X,Y) coordinates.

Color cycling is supported, as well. Since brushes have their own color palettes, you can use SMWG to define (from the sixteen hi-res colors) which colors will be used by each bit map. So, if you want your loaded picture (i.e., a background weathermap) to reserve colors zero through seven, you can assign an incoming brush to reserve colors eight through fifteen. This "double-buffering" of the bit planes limits the color, but that's all you have to work with (until a Hi-res interlace HAM mode is developed).

Perhaps the most impressive feature of the SMWG package is its huge variety of screen wipes and transitions. Forty wipes are available and, according to developer Keith Masavage, more wipes are on the way in additional software support (including the wiping of a brush onto a screen!).

Station Manager Weather Graphics is strictly a presentation software piece that sells for $1,000.00, requires 1.5 M bytes of ram (512K Chip and 1 M of fast ram) and claims to use an overscan (known as FullVideo) of total pixel resolution of 750 X 500 (although the demonstration version did not use this overscan mode). Masavage strongly recommends using Deluxe Paint 2 by Electronic Arts with SMWG. It would be futile to try to write a seperate paint package, since Deluxe Paint 2 covers quite a bit of territory. Together Deluxe Paint and Station Manager Weather Graphics cover two thirds of a complete meteorological workstation! What's missing is the interface to a data base, so that satellite pictures and DIFAX maps can be displayed in IFF format, as well as all the other ASCII text data.

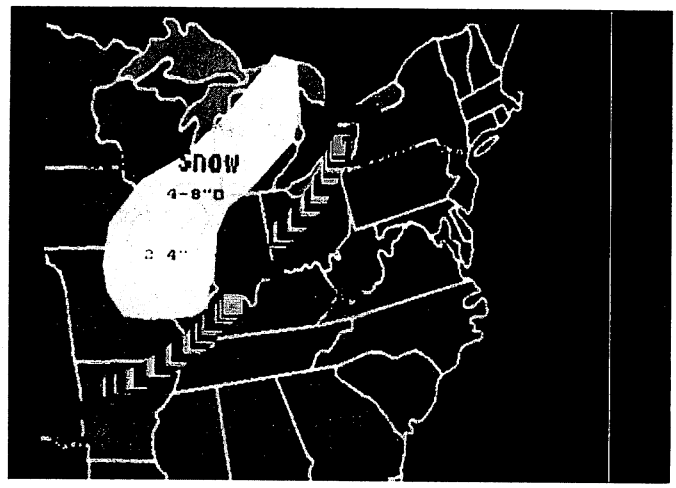
Others

According to some reliable sources, other companies are working on complete weather graphics workstations for the Amiga. Weather Services International (WSI) Inc. is reportedly developing software to allow all weather data, including satellite pictures, to be accessed via an Amiga. No release date has been set.

The same holds true of Accu-Weather Incorporated in State College, Pennsylvania. Accu-Weather is trying to emulate a Tektronix graphics machine, so their unique Advanced Map Plotting System (AMPS) can be made available on the Amiga.

Finally, WeatherBank Inc. is also working on making weather data compatable with the Amiga. All three companies currently support the IBM (with some limits), so transferring the source code from the IBM to the Amiga shouldn't be too difficult.

There is no doubt that the Amiga can provide a television station with an inexpensive graphics workstation and meteorological/forecasting tool. Built-in Amiga™ features, like color cycling, make it a great machine for displaying jet streams and other weather phenomena (see top of page).



John Coleman was quite impressed with the Amiga's capabilities and noted that the machine has great potential. Coleman comes from the day and age when there were no computer graphics...just magnets and markers. He has seen the broadcast meteorology industry expand and is excited about the future. With machines like the Amiga, the cost of high quality graphics workstations will continue to fall and the industry will continue to grow! In the future, when you're watching your favorite TV weatherman, look closely at his graphics and scan the studio for an Amiga!

•AC•

Addresses of Companies Referred To:

Accu-Weather, Inc.
619 West College Avenue
State College, Pennsylvania 16801
(814) 237-0309
Re: Martin Sheridan

Associated Computer Services
1306 E. Sunshine,
Springfield, MO 65804
(417) 887-7373
Re: Keith Masavage

Kavouras Company
6301 34th Avenue South
Minneapolis, Minnesota 55450
(612) 726-9515
Re: Bill Schleeder

WeatherBank, Incorporated.
2185 South 3600 West
Salt Lake City, Utah 84119
(801) 972-5757
Re: Steven Root, Vice President

Weather Services International, Incorporated.
41 North Raod
Bedford, Machussetts 01730
(617) 275-5300

Amazing Reviews...

Deluxe Video 1.2

reviewed by Bob Eller
*People Link AMIGA*BOB*
Delphi 1BOB

A preview of Electronic Arts' latest upgrade
Tips on creating desktop video....

Desktop video is a term Amiga users are hearing lots about these days. Tim Jenison of New Tek, makers of Digi-View, recently defined desktop video as "the ability to produce and manipulate information in video form, without the usual expense associated with traditional video production."

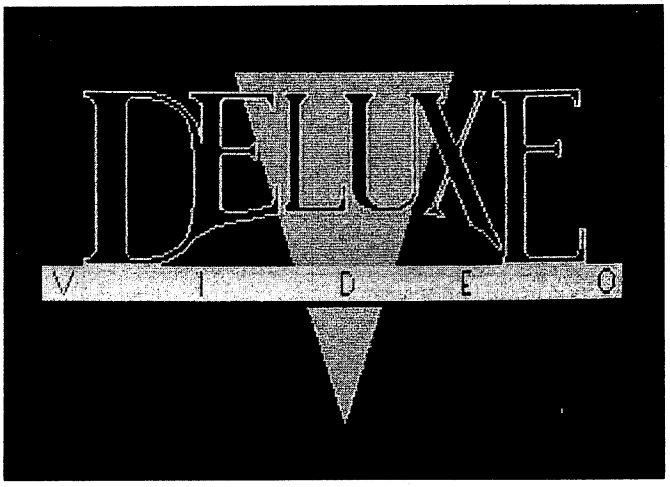
Deluxe Video (DVideo) from Electronic Arts is leading the Amiga video revolution. No other Amiga program has provided the tools needed to bring together the Amiga's incredible graphics, music and sound. Mike Posehn and Tom Casey of Granite Bay Software, authors of DVideo, haven't been content to rest on their laurels. Electronic Arts has released DVideo 1.2 with the new DVideo Post Production Kit.

Users of the original DVideo were dismayed to find that the program had problems when run with KickStart 1.2. The ENHANCER software documenation lists DVideo as one of the programs with 1.2 related problems. I'm happy to report that DVideo 1.2 no longer has these problems. In fact, you *must* run DVideo 1.2 with Kickstart 1.2.

In addition to solving the incompatibility problem, DVideo 1.2 also offers several other welcome enhancements. These improvements include support for the full video mode of Deluxe Paint II and the ability to load music created by Deluxe Music.

According to Electronic Arts' product manager Steve Peterson, "Electronic Arts is committed to improving and supporting all of the Deluxe products. Registered owners will be offered upgrades at an affordable price." If you bought DVideo or any Deluxe product, be sure to send in the registration, so you can be contacted about product enhancements. For information on upgrading your copy of DVideo, contact Electronic Arts at (800) 562-1112.

The DVideo Post Production Kit provides the tools needed to automate the creation of DVideo scenes. The Post Production Kit provides a new set of scene generators (Like the scene generators used to create the scenes with pictures, bar graphs and pie charts already contained in DVideo) to create familiar scenes you've seen on TV and at the movies. These scenes can then be added to your computer videos or recorded onto video tape to create you own TV shows.



Contained on the Post Production Kit disk are pictures, objects, sound effects, music and musical instruments which are automatically assembled into scenes using the new scene generators.

Installing scene generators is a snap with the Kit. DVideo 1.2 contains a new WorkBench drawer, labeled "Libs," which holds the scene generators. To install new scene generators, just copy each generator from their drawer on the Post Production Kit to the Libs drawer on the DVideo "Maker" disk. Installing scene generators, in this way, frees your Maker disk for other video parts. You might, for example, create a special Maker disk for each of the scene generators, including only the video parts needed for a particular video. If you have plenty of room on your Maker disk, the Kit includes a program to automatically install all the scene generators to the Maker disk Libs drawer.

Once you've installed the new scene generators on the Maker disk, you'll have four new menu selections available for creating scenes. 'Classics' contains a sub-menu listing parodies of familiar Movie and TV scenes. 'TV News' provides a selection of scenes fromTV news. 'Titles' combines graphics with your own titles. 'Wipes' is an interesting way of moving from one scene to the next in your video. Let's take a closer look at each.

continued...

Classics

We've all seen the flashy openings producers use to grab audience attention. With Classics, you can add a touch of Cecile B. DeMille to your video. The Post Production Kit includes seven familiar, eye-catching openings and the accompanying music. These Hollywood-style openers include:

20th Century Pix - The picture fades in and searchlights blaze across the screen. As the picture appears and the fanfare plays, your friends will think you've taken a job in Hollywood.

Pinnacle - Pinnacle is another familiar movie opening. The clouds move past the mountain tops and the song *Pinnacle* accompanies your opening.

Liberty Pictures - Americas' favorite lady, the Statue of Liberty, holds the glowing torch, signaling the start of the production. The torch is created using animated cel objects.

Amazing Videos - One of my personal favorites, Amazing Videos uses animation to build the word "AMAZING" and completes the scene by fading a more detailed picture behind the animated text.

Twinkle Zone - This is a fairly simple black and white picture that fades in and out. The accompanying music makes this one special. Further custom credits can be created in the same black and white style and played following the opening.

Amiga Vice - You may have seen the art for this scene in the recent issue of Electronic Arts' Deluxe News. If you are lucky enough to have a genlock, you can overlay this scene on a home video - Crockett and Tubbs got nothing on you! The music for this scene is also well done and can be played after the scene, while rolling your own credits.

In Color Stereo - Move over NBC! As the scene opens, a colorful turkey flaunts his multi-color tail and the network stereo headphone logo moves to the bottom of the screen. Color and stereo, your Amiga's got 'em and you can tell the world!

Titles

Each of the scene generators in the Titles sub-menu allows you to combine graphics from the Post Production Kit with your own text. The ease of creating custom titles text really highlights the advantage of scene generators. A polished custom product is created with very little work. The Post Production Kit offers six scenes to display titles, including:

Theater - While creating the scene, the generator asks you for three lines of text. Each line can be up to eighteen characters long. The scene displays your text centered on a movie theater marquee.

Marquee - How would you like to have eighteen characters to display your credits? Marquee allows six lines of twenty-five characters each! Your message is displayed on a large marquee, surrounded by flashing colored lights.

Flashy Text - Want to grab your viewer's attention? Flashy Text does the job. Two lines (up to fifteen characters each) are displayed in a yellow box with black letters. The box grows to fill the screen and the yellow and black letters alternate for an extremely flashy effect.

Blimp - Another favorite. The GoodStuff blimp floats by, displaying 4 lines of text in lights at the base of the blimp.

Silent Movie - This scene uses the Amiga's Emerald Font (twenty point) to display six lines of text (up to twenty-one characters long). The text is centered in an ornate border and looks as if it were taken from a silent movie.

Fading Text - Every video production needs to credit those who participated in its creation. Fading Text allows you to fade eight lines of text in and out. If you want several lines of text to appear at the same time, a code can hold your display. Need more than eight lines to display your credits? Just generate several scenes with Fading Text and have them follow each other in your video.

TV News

Bad news got you down? No problem, just create your own news! TV News provides the generators for creating your own news program complete with weather and sports. Who knows, you may get an offer from a network! In addition to

the available scenes, Post Production Kit includes several objects that appear on screen with your newscaster, through Genlock, including:

News Intro - A simple, but elegant introduction to your news program. Follow this intro with a credit scene, listing your news team.

Sports Scores - You can input seven lines of text (up to thirty-one characters per line) listing the league and scores. The league name is centered and underlined at the top of the scene with the team scores paired below. The scene generator formats your display and uses a wait effect. When your newscaster gives a cue, hit the space bar to end the display.

Current Weather - This one should look familiar. Enter the temperature, humidity, pressure, wind, visibility and rainfall. . . and the current weather is displayed. This scene also includes a wait effect.

Extended Forecast - This scene shows a four-day forecast, complete with a graphic representation of each day's weather. Codes represent most weather conditions, including tornadoes. By drawing a map of your area, you can create a custom weather scene, using these weather symbols and your map.

Wipes

Wipes provide an interesting way of moving from one scene to the next in video work. In DVideo, wipes move the foreground picture in front of the background and then wipe the foreground away to reveal the background picture. The Post Production Kit gives you four wipes for special effects work, including:

Flames - Flames lick the bottom of your scene and gradually move towards the top. As the Flame spreads, your background is uncovered, bottom to top.

Spirals, Random and Zigzag - These wipes are essentially the same as Flames, only with a different wiping pattern. For more personalized wipes, you can generate the wipe and modify how the wipe moves and the pattern it follows. For a great special effect, leave the masks created by the scene in place, and leave the foreground strobe off. As your wiper moves around the screen, you have the effect of looking at the scene below, through a moving window! Getting some more ideas about how to use a Genlock??

Hints for better videos

Much like normal video production, desktop video production involves three stages. Pre-production involves deciding what your video is about, planning each scene and creating the art work needed to convey your ideas. The production phase of video work means taking the pre-production work and putting it together in a working video. In post production, you fine tune the timing of your video, add the special effects and credits and polish your work.

Putting together animation, graphics and sound consumes lots of memory. Fortunately, the Amiga has plenty of room for memory expansion. I recommend you expand your memory, if desktop video work is part of your plans.

While you can create videos with a 512K Amiga and one disk drive, production is much easier with an expanded memory and a second drive. Two megabytes of added memory let you load all video parts to memory and create and run the video without slowing down to fetch data from the disks. Playing the video from memory smooths out the playback and gives your product a more professional look.

The video tape equipment you select for your video work is also an important consideration. Consumer decks are available to add professional effects, such as painted mosaics and wipes. The ability to tape individual frames on your tape deck allows you to run DVideo in single frame mode, creating smooth, professional animated videos. When you are in the market for a new piece of equipment, go for the highest quality product with the most features within your budget.

Desktop video is truly one of the new frontiers opened by the capabilities of the Amiga. I'd be happy to answer any questions you have about video and the Amiga. Contact me on an electronic network or drop me a line at *Amazing Computing*. The more we share our knowledge, the better our video work becomes. . . So keep that production rolling!

•AC•

Amazing Reviews...

Pro Video CG1

*A Character Generation Program by Jeff Karline of **Pro Video, JDK Images***

*Reviewed by **Oran Sands III***

CG1 cannot be easily plugged into an existing pigeonhole of computer programs—It's a very single-minded program. CG1 makes the Amiga imitate a broadcast video character generator. Nothing more, nothing less. Don't try to evaluate this program as anything else. This narrow view of things helps the program excel at its intended purpose. It is intended for the video professional (but it's also simple and inexpensive enough for the video amateur). CG1 allows you to replace most character generators in small television studios, industrial (or educational) studios and cable systems with an Amiga. . .at a very reasonable price!

A character generator (c.g.) allows the video professional to place text information on a tv screen in different fonts, colors and sizes. The c.g. is usually in use when screens of statistics are displayed (in news and sports broadcasts, for example). CG1 by Jeff Karline of Pro Video, JDK Images is a sophisticated, yet simple piece of software that transforms the Amiga into a character generator.

The program was written in machine language to enable the Amiga to utilize all available memory and functions. CG1 does not take advantage of the Intuition interface, as there wasn't enough memory available in a 512K machine for both the program and Intuition. In fact, early versions of the program worked only with Kickstart 1.1 and couldn't see external memory. Since the program was so memory-intensive, any external disk drives had to be disconnected, so the operating system wouldn't allocate any buffer space for the drives. All in all, the program takes all but 1500 bytes of the available 512K! Why so much ? Because Jeff found a way to keep 100 high-res screens in memory at one time!!!

Features

When using a c.g., you are concerned with many individual pages, as well as sequencing groups of pages. CG1 addresses these needs quite efficiently. The options for designing a page are overwhelming. Let's try to list these features:

Variable by character.... text color, font, text size, character/space underline on/off

Variable by line.... underline color, line size, background color, background grid style, size and color, character outline/shadow style,size and color

Variable by page.... color palette (8 of 4096), color flash on/off, alternate character fonts (3 fonts— each in 3 sizes, always resident), transition style, speed and page dwell time.

With these options, it's not uncommon to spend 20 minutes designing the first page, and the next 20 minutes doing all the rest of the pages.

Ease of Use

CG1's looseleaf bound manual is quite clear. The early chapters walk you through creating your first page of text. Appropriate cautionary notes are provided.

A cutout template is provided for the function keys which are used for almost all operations. The function keys bring up boxes offering selections that are chosen by the cursor, up, down, right and left keys and, on occasion, the + and - keys. As you select the different choices, the font and its particulars inside the box change to echo your choice— A slightly different version of WYSIWYG. Call it "what you see is what you WILL get."

All selection boxes can be exited at any time by pressing 'Return'. Your choices remain active until changed. All color choices are made from the currently active palette. The active palette is picked from 15 preselected palettes, or you

may chose to modify the existing one. All 4096 colors are initially available. The resulting palette consists of eight colors, the first of which is the screen background (and border) color (Color #00 for the curious).

EDIT MODE

Entering text is as simple as typing with a word processor. In fact, the entire page could be typed and then you could reposition the cursor and begin choosing colors, shadowing, etc. While you work in edit mode, the screen is displayed exactly as it is currently designed (color, font choice and all). While working at the bottom of the page, any selection boxes that appear, suddenly switch to the top of the screen, as not to obstruct the line you're working on. Your edit decisions are always right before your eyes.

While in the edit mode, every function is up for grabs at anytime. Nothing is etched in stone until saved to disk. After designing a particularly tasteful page of text, you may want to recreate that style on the next few pages. A page duplication feature allows you to do just that. Just reenter the new page's text to have two different pages of matching information.

Once you have more than one page of text, you'll need to decide how to make transitions between them. 15 choices are available: fade ins/outs, push on new page, push off old page, reveal new page under previous page, go to new page one piece at a time via a checkerboard effect. . . it's too difficult to describe all the selections, you have to see them for yoourself (Check end of review for Demo details).

The manner in which you transition can make your pages update the text, without losing the original. The speed of the transition can be varied in seconds and the dwell time (which includes the transition speed) of the page on the screen can be precisely determined. Once these choices have been made, it's time to go into PAGE mode by pressing the ESC key.

PAGE MODE

Page mode allows you to select pages for display (or output). No selection boxes pop up to mar the picture. In fact, all functions are now handled via the function and cursor keys.

Press F1 and pages 00-09 will sequence as designed (pressing F2 accesses pages 10-19, F3 pages 20-29, etc thru page 99). Undefined pages are passed over (any page can have its transition defined as "pass," if desired) and the next active page is shown. Pressing F1 and the Shift key together will begin the sequence of pages 00-09, but will not stop at the end of the sequence. Rather, it will repeat all selected pages until you stop it yourself.

This feature is handy for repeating messages, much like messages used on cable systems to fill unused channels or lag time between programs. Groups of pages (in tens) can be set to cycle, allowing more than just 10 pages to be cycled. If, at any time, you notice that a page doesn't dwell long enough or that something is misspelled, you can always exit page mode and reenter edit mode by pressing the ESC key. Just press ESC again to reenter page mode.

ODDS, ENDS AND OTHER THINGS

Jeff Karline is an ex-cable tv producer and understands the valuable uses of CG1. Therefore, he took the time to address a few problems the Amiga has with video. Pressing CTRL-C will re-time the horizontal position of the screen, so that the composite video output of the Amiga will be properly centered (this is usually done thru Preferences, but with Intuition gone, you must find another way). RGB output users won't need this. It's also no problem to use a genlock device with the Amiga— it was designed to work with genlocks from the beginning!

The latest version of CG1 allows Kickstart 1.2, external memory and DF1: to be used together. This expansion gives you no more features, but does allow you to access the program from DF0: and default to DF1: for the data disk. The original version of the program using Kickstart 1.1, one drive and 512K is also on the disk, just in case you don't have a second drive and external memory.

The program allows three fonts, in three sizes, to be resident at all times. The fonts supplied with CG1 are quite good and have been chosen to match those most often used by stations

continued...

when titling. Loading a new font removes one of the fonts from memory. 4 new fonts are now available for CG1 and 4 more should be on the horizon by the time you read this article. The total number of choices is now up to 11 (counting the three that come with the program).

Originally, these fonts weren't compatible with Workbench fonts and programs utilizing the Workbench fonts. I'm pleased to say that these fonts are now available on a separate disk (all eleven), in a compatible file format for Workbench fonts. These hi-res fonts are simply better than any I've seen to date and the large size (80 lines tall) is splendid! The attention paid to shadowing and the design of fonts virtually eliminates any hi-res "flicker" on *any* combination of colors or characters.

In lieu of entering text, the keyboard can also be used to enter graphics characters by pressing the ALT key in conjunction with the desired key. A keyboard map is included to show which characters are available. Boxes, lines and stars can be used to attract attention to your text.

The only bug I could find in this program was in disk handling. Since all disk drive error messages are handled thru Intuition routines, it is not possible for the program to pass on these messages thru the usual requestor boxes.

I hung up the program by saving my job (all 100 pages are saved or loaded at one time); the program wouldn't acknowledge the program disk. It was a small problem, attributed to the fact that I had write-protected my backup copy of CG1 (which is NOT copy protected) and that's a no-no. Nothing was lost and a warm boot and reload returned me to normal. If I had followed the manual's guidelines correctly, I wouldn't have had a problem!

Jeff tells me there wasn't enough memory to allow for better message passing. Since disk operations are limited to booting, loading a job (the first thing you might do) and saving a job (the last thing you'll do), you'll probably have only three or four disk operations per work session. I doubt you'll run into problems.

The graphics are obviously NOT in IFF standard format. Accordingly, you cannot get 100 high-res (640x400) pages on a disk. Hopefully, future versions of the program will include a "bridge" between IFF and CG1. Don't expect to combine these graphics with Deluxe Paint images, but do expect to find yourself putting professional-looking text on your video presentations!

The only caveat I have about this program is that the resolution isn't as good as a Chyron-type ($30K and up) character generator. On the plus side, though, the res is perfectly acceptable for non-broadcast work (which is 70% of the tv work done today!). Scrolling text up the screen isn't possible, but the choices of transitions allow for more creative and better looking methods of displaying text. Many expansions are being discussed and evaluated as part of CG2, which is under development at this time.

SUMMARY

The program is supplied on one disk; a master data disk is supplied on another, and both are bound in a vinyl 8 1/2"x 11" 3-ring binder which also contains the loose- leaf manual (41 pages, including graphics keyboard map and function key templates). Telephone support has been excellent and quite informative. Demo versions of the program are circulating the user groups and are available for free from JDK Images.

•AC•

Pro Video CG1 $199.95

JDK Images
2224 E. 86th St.
Bloomington, MN 55420
(612)-854-7793

Additional fonts $34.95 ea. or $99.95 for 4

AMAZING REVIEWS...

Digi-View 2.0

"Great video stuff that's easy to learn."

Reviewed by ***Jennifer M. Jarik***

Those lazy summer days are here. You venture out, with your thirty-five millimeter camera, in search of great pictures. You capture some memorable images. Now you'd like to capture the haze of that summer day with your computer. ***Digi-View 2.0*** is the answer.

Digi-View is a video digitizer. In combination with a video camera, Digi-View hardware and software captures the image and displays it on the screen. The images can be saved as IFF files. Once digitized, Digi-View images can be manipulated in many ways. You can transfer the images to paint programs, print them out or send them to anyone who has a modem. (For more about Digi-View, see Ed Jakober's review in *Amazing Computing* Volume 2, Number 1.)

Digi-View 2.0 is an upgrade of the original Digi-View software. The original Digi-View allows you to digitize in five resolutions: 320 by 200 low res, 320 by 400 low res interlace, 640 by 200 medium resolution, 640 by 400 high res color and 640 by 400 high res black and white. High res color and interlace modes have been added to the new version. The higher resolutions do require expansion memory.

As well as digitizing camera images, the Digi-View software is also a powerful tool in itself. Digi-View 2.0 functions without the digitizer attached to the Amiga and truly compliments high-power painting tools, such as Deluxe Paint II. You can load existing IFF pictures and change them with the color and palette controls, as if the images were freshly digitized. The new software also converts images between resolutions. A low resolution HAM image can be loaded into a high resolution sixteen color mode and displayed in correct proportion and sixteen flashy colors.

Many of the new features lie under a command menu titled "Control." "Color Control," "Color Palette" and "Camera Control" are especially useful.

"Color Control" has seven slider bars and various gadgets. The "Mode" gadgets at top of the screen select the display mode. Your choices include black-and-white, the current number of colors (usually 32 or 16), 4096 for ordinary HAM pictures and a new mode, 4096+. Black-and-white is nice to have only if you have a black-and-white printer.

The Digi-View manual chimes that 4096+ produces "the best possible image". The 4096+ mode translates a more life-like image; the clarity is still there, yet the colors blend into each other. Compared to my original photograph, the Digi-View hazy summer day picture seems real! The new HAM mode sharpens the image and prevents the HAM "bleed," noticeable at the edges of sharp color transitions. The manual does advie that you hold off on 4096+ until you've got the other control settings exactly how you want them.

Another useful mode setting is "Dither." Dithering takes the color of two adjacent pixels and finds an intermediate color. When I first used this mode, I didn't see any difference in the image . A closer look at the line by line translation of the image, though, showed much color variation.

Interestingly enough, the further away you are from the screen, the more impressive the effect of dithering. This improvement with distance resembles the painting technique "pointillism." With pointillism, small dots of pure color are applied to the canvas with the point of a small brush and the effect is much more dramatic from a distance. Dithering forms a picture using few colors, by placing dots of color side by side, giving the impression of a third blended color. There are two "Dither" commands: "Dither 1" and "Dither 2." "Dither 1" seems to be about half as dithered as "Dither 2."

Alongside "Dither" is the "Pos/Neg" Control. This command makes a negative of your image. For color pictures, "Pos/Neg" replaces each color with its opposite on the color wheel. An orange turns blue, a red turns green, etc. Saturation and brightness are similar, but the hues contrast. For black-and-white pictures, the negative resembles a black and white photo negative.

"Brightness" is the first slider control, similar to a televison brightness control. All colors on the screen can be made brighter.

"Contrast" (my personal favorite) is the second slider control. As you move the slider up, the whites get whiter and the blacks get blacker. As you move the slider down, your image becomes more and more gray.

"Saturation" is the third slider control. When pushed all the way up, the colors make you feel like you're watching a bad cartoon. The more you move the slider down, the less

continued...

intense the colors. This type of control is nice if your image is over saturated. You can tone an image down, print it out and hang it in your living room, without your eyes aching.

"Red, Green and Blue" are the next three slider bars. These slider bars lend an overly reddish, greenish or blueish cast to your images. If you want an image to look sunny, give it a blueish cast. Green is nice for southwestern images. Red brings on a sunset feeling.

"Sharpness" is the last slider. Raising this slider gives your picture a grainy or snowy look. Dropping this slider blends colors. One of my favorite tricks is combining "Sharpness" with "Dither". When changes in sharpness make the image too grainy, I dither the image to soften the sharp changes. Conversely, when the image looks a little too dithered, I lower the sharpness and the colors blend together more smoothly.

Many other gadgets help make up the color control requester. "Default" returns the slider bars back to the center zero position, allowing you to correct your boo-boos and start again. "Display" recalculates the picture according to the current settings. "Camera" moves you to the camera control requester. "Palette" switches to the palette control requester.

There can be some debate about the greatest feature of Digi-View 2.0: the color control or the new palette abilities. "Artsy" types enjoy playing with the color control because of the saturation, contrast, sharpness and dither commands. "Programmer types" may favor the palette. My opinion is that color is easiest to control with palette. . . But the variation for the highly color sensitive takes place in the color control requester. Hmmm.

The palette requester shows the current palette. After the Digi-View digitizer has captured an image from the camera, it chooses a palette to represent the picture. Digi-View also selects a palette when converting a HAM picture to non-HAM mode. If you disagree with the computer's "eye," you can select a color and change it with the RGB color sliders.

You can choose how many colors you'll use with another slider. You can recreate a picture in thirteen colors, for example. This feature enables you to transfer images to other resolutions in paint programs. Impressive.

You can also create a half-tone image from a color picture with Digi-View. Turn up the contrast and brightness, reduce the palette to two colors and set those colors to pure black and white. The dithering algorithm in Digi-View is much smarter than the algorithm used in PageSetter, so you can use Digi-View software to create nice halftones that print very well on a black-and-white printer.

A nice feature of the palette requester is "Color 0." "Color 0" is Amiga technical language for no color. If you select a color you wish to set at zero, like magic it disappears (It actually becomes the transparent color.). If you don't want holes in your digitized picture, remember to turn off "Color 0" when you aren't using it.

The palette requester also includes mutually exclusive "Freeze Palette" and "Make New Palette" options. Freezing a palette tells the program to keep the current palette when it displays the next image, instead of choosing a new palette. This option lets you load a palette from another IFF picture, while allowing Digi-View to remap the colors of another picture into the loaded and frozen palette. In this way, you can digitize images with identical palettes, then clip brushes between palettes, without the usual hassles of remapping and restoring brush palettes.

One of my favorite tricks is to save a palette and then use "Pos/Neg" to find the best contrasting colors. I write down the RGB values for a few colors and retrieve the saved palette. By raising the contrast and modifying the colors that are similar to the contrasting colors found in the negative image, a highly contrasting, very dramatic image can be created.

The most interesting feature of "Camera Control" is its control of scanning time. If you want an idea of what an image will look like, you can scan a picture in only five seconds. The normal ten-second scan is also available. The new software has greatly improved Digd-View's ability to get good pictures from a color camera. A twenty -second scan produces true-to-life colors with color cameras and removes the streaks and banding.

My only negative comment about Digi-View 2.0 is that truly professional video needs true overscan to remove borders from pictures. Digi-View really needs to produce images that cover all edges of the screen, so no colored borders remain.

Otherwise, Digi-View 2.0 has many, many good points. . . It's an achievement in itself. Even the manual is well-written, making Digi-View a pleasure to learn. Great video stuff that's easy to learn—a great feat well accomplished.

•AC•

Digi-View 2.0 $199.95
Upgrade cost $10

New Tek
701 Jackson Suite B3
Topeka, KS 66603
(913) 354-9332

Amazing Reviews...

Prism HAM Editor

from Impulse

Put ALL the Amiga's famed 4096 colors to use!

*Reviewed by **Jennifer M. Jarik***

You've fiddled endlessly with the lights, patiently moved the camera to just the right angle and created a near perfect HAM picture with the Digi-View digitizer. Your almost perfect HAM picture could use a little change of color, though, a little clarity here and there. With this sort of problem in mind, Impulse created a HAM editor called Prism.

HAM pictures are the multi-color pictures you can create with Digi-View. All the Amiga's famed 4096 colors are put to use. HAMs also follow certain IFF standards. Prism allows you to alter colors and draw on your HAM picture. Prism tries very hard to imitate and does much of what D-Paint can do with thirty-two color low-resolution pictures

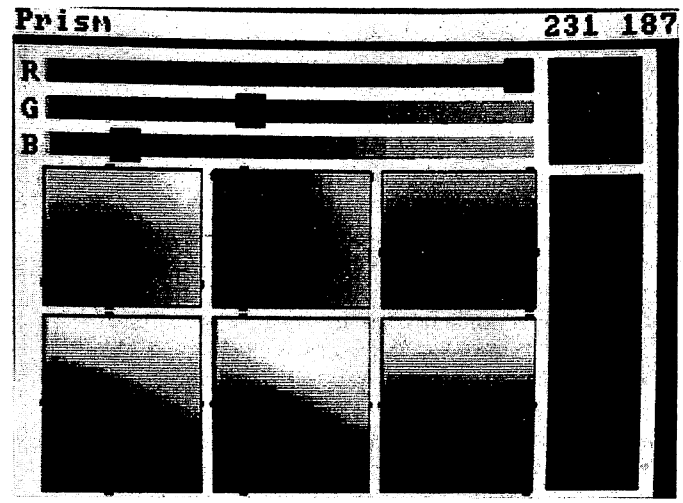
Prism requires 512K and, like Deluxe Paint, has a toolbar along the right edge of the screen (the toolbar can be hidden if you wish). Fourteen built-in brushes appear along the top of the toolbar, including various sizes of circles, squares and spray paint brushes.

Colors, Colors, Colors

A rainbow-like icon, beneath the brushes, brings up the 4096 color palette requester. Six multi-colored squares immediately catch your eye. These squares access all 4096 colors. When you click in the squares to select a color, all the colors in the squares change.

When I first started using this part of the program, I wanted to change a blue to a red on my HAM picture. When I found the red I wanted, all six squares became variations of red. The best of blues disappeared. The colors change in a uniform pattern. On the top half of the squares, the colors change as tints (colors combined with white). The bottom squares change in shades (colors combined with black).

Above the six main squares are three slider bars labelled R, G and B. The manual notes that the sliders are "used for fine-tuning" the six main color squares. I found a brown that I sort of liked and tried to change it with the slider bars. As I moved the slider bar, the colors inside the squares changed, but nothing was fine-tuned. The colors in the final six squares were radically different from the colors in the original six squares.



The most confusing element of Prism is that the color bars themselves changed from red, green and blue to all sorts of unrecognizable colors. Perseverance and many cups of coffee helped me discover the cyan, magenta and yellow components within the slider bars.

Sliders

When the sliders are at the left, the colors in the six main squares are tints of red, green and blue. As you move the sliders to the right, the colors in the squares become tints of cyan, magneta and yellow. Perhaps Prism should add a tutorial to the manual explaining the connection between the color squares, color separation and color sliders.

After wandering through the color finding maze, I found a way to keep the red, blue and brown I liked in my color palette, so I could paint with them later. Just click the pointer on the color you want to keep. At the upper right, in the palette requester, is a box marked "OK". This requester also has an area that holds a palette of sixteen colors of your choice. You simply click on a color to retrieve it. If you try to put more than sixteen colors in the palette, the first color you saved is tossed out.

continued...

If you'd like to display your "perfect" HAM pictures, Prism provides true cyan, magenta and yellow color separations. Color printers use a combination of these three colors to create all colors on paper.

The Manual

The very first page of the Prism manual - in fact, the very first line - reads, "If you don't read manuals...it's OK (sort of)." The manual goes on to say that if you had no trouble running Deluxe Paint or Aegis Images, then you'll have no trouble running Prism.

Well, I know how to run both D-Paint and Images precisely——in fact, my business requires such knowledge—— and I had more than enough trouble wading through the maze of the color palette. How about users who want all the colors Prism can provide, but have little experience with paint programs? Or users who've just bought an Amiga and know very little about it? All these people would be totally lost.

The Prism manual also gives a detailed explanation of exactly what a HAM picture is, along with hardware requirements and a picture of the toolbar. The next five pages give one- and two-line descriptions of each Prism command. . . but no direction at all on how to make these commands work!

Nearly half the commands refer to "copy", which is later foggily defined as "similar to a brush, but you can't paint with it." Much fuss is also made about "regions," which the manual doesn't highlight until two sections later. Between bouncing from the references to the unknown "copy" and "regions," the brief descriptions are even more confusing than no descriptions at all. The next two pages dumped the keyboard equivalents to the commands I didn't understand from the previous pages.

Then, finally, I struck some gold in the Prism manual. A brief description of how to use fonts and the 4096 color palette, and a definition of "copy" and how to access it, all proved quite helpful.

Project Menu

The first pulldown menu is the project menu, including all the usual commands: "Open" (equivalent to "Load"), "New" (equivalent to "Clear"), Print and Quit. "Save as" allows you to save a picture in IFF form, as either IFF RGB4 or IFF ILBM. "ILBM" accesses previously saved IFF HAM pictures from other programs, such as Digi-View. "RGB4" is a unique save mode which uses less memory than usual IFF forms.

Region Menu

The next pulldown menu is the region menu. Prism can maintain two pictures at the same time, one on the screen and one in the undo buffer. Prism supports a host of commands operating between the two pictures. If these commands are used incorrectly, you may think you're changing the picture on the screen, when you are actually changing the picture in the buffer. Sort of confusing.

Five main commands relate to the regions. You are working with the picture in the undo buffer when you do a "Copy" region or a "Restore" region. "Keep" region lets you work with the picture currently on the screen. "Swap" region exchanges the two pictures - the one you see on screen is placed in the undo buffer and the one in the buffer pops onto the screen.

To save the changes you've made on a picture, you have to "Lock" them in. . . if you forget to turn on "Lock," you lose all the changes you've made. None of the changes will be there when you try to "Keep' region or "Swap" region. Sort of frustrating.

The Copy Command

The "Copy" command is quite interesting. "Copy" allows you to take any portion of a HAM picture and copy it onto another spot in the picture. Unlike the brush commands in Aegis Images and Deluxe Paint, "Copy" does not make long, continuous versions of the image; it simply copies one portion of a picture onto another portion. Watch it, though, if you don't change the regions properly, you may end up copying onto the picture in the undo buffer.

The first step towards using "Copy" is activating "Lock" in the Modify menu. Pick your brush from the toolbar, outline the section of the picture you want to "copy" and press the F1 key. The manual states that an icon labeled "Pick" should appear at this point. My icon came up as "Fill." Puzzling. Apparently, I was to choose whether I wanted to "Copy" everything inside my enclosed area or everything outside the area. After struggling with my discrepancy, I decided "Pick" and "Fill" must be the same thing. Mysteriously, my "Copy" was complete.

You can create some interesting effects with Prism's last menu, the Display menu. You can display all the colors in the HAM mode or you can display just the red, green and blue layers of a picture!

Gripes

Prism creates a rather odd-sized IFF color map hunk. This quirk mushrooms into a significant compatibility problem. These IFF HAM files created by Prism do not load into the Digi-View 2.0 software. The Liquid Light Polaroid screen printer also does not recognize these Prism files. Impulse's Stan Kalisher counters that an update to Prism is on the way, correcting this problem. In the meantime, Kalisher recommends public domain programs such as the SaveILBM and Discovery Software's Grabbit to save an IFF file that Digi-View and Liquid Light recognize.

An Overview

Some overall conclusions about Prism. Prism is a useful program, if you have the time and patience to wade through it. The similarity to Deluxe Paint and Aegis Images simplifies Prism somewhat, but the lack of tutorials and "Amiga" terminology makes Prism generally difficult to learn. The price, 69.95, is reasonable, but be prepared to tackle the problems.

•AC•

A-Squared and the Live! Video Digitizer

by John Foust

In the Amiga museum of Products That Never Showed Up, the Live! video digitizer has held a central position. That may change in the near future, according to representatives of A-Squared and Grab, Inc., a new company that will bring Live! to market. It may appear as early as August 1987, priced at $295.

Live! was first demonstrated, at the launch of the Amiga, in the summer of 1985. At the launch, artist Andy Warhol used it to capture an image of rock star Deborah Harry. Live! is a real-time video digitizer. Given the image of a living, moving actor from a video camera, it can reproduce the image on the Amiga screen. The image can be captured instantly, like a traditional camera, and saved to disk for editing with ordinary paint programs.

Live! was designed by Arthur Abraham and Wendy Peterson at A-Squared in Oakland, California. Commodore agreed to market the device for A-Squared under the Commodore name. A long series of delays have prevented Live! from being sold. The delays can be attributed to several sources, both at Commodore and A-Squared, and negotiations between the two companies.

In May 1987, Grab, Inc. negotiated with Commodore to get the rights to distribute Live!. Before this, Commodore maintained they would market the device themselves. Now, most parties involved acknowledge Commodore will never market Live!. Commodore planned to call it Amiga Live!. With the advent of Grab, Inc., it will be called simply Live!, to avoid conflicts with the Commodore trademark.

Live! history

There is a long and complicated history behind the delays of Live!. It all started with a few meetings with Amiga. According to Wendy Peterson of A-Squared, "We worked very exclusively with Amiga. We met them first in September 1984, a week after they had been purchased by Commodore. They were a real good group; we wanted to work with them."

In September of 1984, A-Squared had no hardware to demonstrate. They got a "black box" prototype Amiga in January 1985, after creating a proposal for a product. Amiga engineers had thought about video digiziters, but they didn't think of a fast digitizer, only a slow digitizer.

"We got it up and running in May 1985, so they put us in the Amiga launch in July," said Peterson. "At that point, Commodore started talking to us about picking it up. Before then, Amiga suggested to us that Commodore might pick it up. We didn't start out with any assumption that something like that would happen. We were thinking about doing it ourselves."

"Live! had a number of champions around the time of the launch. Commodore didn't really come into the picture until the launch happened. At the time, Frank Leonardi loved the product. Clive Smith was interested in a lot of vertical market applications. But it was mostly a thing between A-Squared and Amiga."

They thought a contract with Commodore would be better than doing it themselves. The four-layer design of the board was beyond the abilities of A-Squared. Peterson continues, "So we decided that since Commodore was very interested. and they could make it available to all the retail outlets, where the Amiga was sitting, side-by-side with Live!. We thought it made sense. We could spend our time writing software and selling to an installed base."

"We negotiated an agreement over a period of many months and finally signed it in January 1986. A couple months later, we signed off [preliminary designs of the board.]" Amiga and A-Squared moved ahead and laid out a circuit board for Live!. About this time, in the spring of 1986, the first of the Commodore engineering layoffs delayed Live!. Support for the product also diminished among Commodore upper management and some supportive managers left.

When new management arrived, they had never heard of Live!. Peterson explains what they saw: "At that point, all they saw was Digi-View, which on the market and beginning

continued...

to sell well. When it first went out, it was sort of a stopgap kind of product. The software came up really quickly and Tim [Jenison, of NewTek, the makers of Digi-View] has done some excellent image processing software, so now it is a product in its own right."

Their unfamiliarity with history, combined with the layoffs, slowed Live even more. "That is what the new people at Commodore saw. They were saying, 'Digi-View is great! What have we got? This A-Squared [box] only seems to be problems! We've wanted to get it going for half a year now and nothing is happening.' Of course, that was because there wasn't anybody around to get it out. The engineers at Amiga who picked up Live! would periodically leave. Every time a new engineer came along, they would have to relearn it and then they'd go away."

By January of 1987, Live! was submitted for FCC clearance. At the same time, the original contract with Commodore was due. It was a year-long licensing agreement, involving a license fee and royalties. Commodore was anxious to rework the contract because of the wording of a single clause.

Peterson described the contract clause quite diplomatically. "It had some phrases that appeared to work to their disadvantage." The contract said the rights to market Live! could revert to A-Squared, if Commodore had not sold a certain number of units after a year. The contract mistakenly measured the year from the contract signing, instead of when sales commenced so the rights to Live! could have reverted to A-Squared without a single unit being sold.

At the January Consumer Electronics Show, A-Squared and Commodore worked out a new contract. Peterson also wanted to see Live! demonstrated in the Commodore booth. "I said, 'Come on, you guys, let's get this thing going.' I want to see the product down there being demonstrated. When a customer comes up to Commodore and asks them about the product, I don't want them to laugh in their faces. I want them to answer in an intelligent way, when the product is coming out, or talk to them about its capabilities. Let's pretend that this is real."

Why would Commodore delay Live! for so long? There was a slim hope that the product would make it to market eventually, but no one got behind it and pushed. According to a source close to Commodore, Commodore chose to sit on Live! out of embarassment. If they were to lose Live!, this reasoning goes, then they would have lost all of the planned third-party products for the Amiga. These products include CalcCraft, MusicCraft, MovieCraft and TeleCraft, all announced at the Amiga launch. They invested a lot of money on these products and then lost each in turn. They didn't want to lose the technology of Live!, but didn't have the drive or the resources to bring it to market.

Final chapter

This announcement might seem like a rerun, but the Live! digitizer is finally coming to market. The Live! system will be distributed independently of Commodore, by a company called Grab, Inc. The husband and wife team of former Commodore-Amiga programmer RJ Mical and present Commodore-Amiga employee Caryn Mical founded Grab to sell the Live! board.

"Our primary chapter is to bring this product to market," according to RJ. "I always thought Live! was a great product. This is the second time I've snatched Live from the jaws of doom." The first time happened when A-Squared first demonstrated the Live! board to Commodore and Amiga. "That initial interest lasted for about a year and half. Now, it's died again."

Commodore has decided not to make Live!. The Live! board is not currently being manufactured, contrary to a popular rumor that said it was in production and would be sold this spring.

"If everything goes wrong, we'll ship the product in three months. If it all goes right, it could be five to six weeks," Mical said in late May. Mical will assist Abraham with the Live! software. In particular, Mical will improve the user interface.

Mical is confident the Live! board will be available soon. Parts are on order and the board manufacturer is gearing up for production. "I've brought in experts to review the whole thing: reviewing the product, the parts, the purchasing, the manufacturing process. I brought in some of the old Amiga people for a critical review. After their critical review, some of them chose to become investors. There is nothing that can stop it at this point. It's a great toy. We're planning on selling to people who like great toys - high tech children."

The Live board is an external expansion box for the side of the Amiga 1000. It is a "bus end;" it does not pass the bus, so no other devices can be placed on the other side of Live!. Live will come with one piece of software, a program to capture images and adjust the characteristics of the board.

Grab is negotiating with Commodore for the rights to a version for the Amiga 500. In the future, they hope to have more advanced software, such as a video special effects package.

Live! will be available only through mail order. For ordering information on Live!, call Grab Inc. at (800) 626-9541, extension 1156. A-Squared Systems is at 10 Skyway Lane, Oakland, CA 94619, phone (415) 633-0703.

The future of Live! remains uncertain, according to other cynical sources involved in the Amiga video market. Other companies are developing rival video products; precisely because of the delays in the arrival of Live!. There will be more video digitizers in the future and they may influence the ultimate popularity of Live!.

•AC•

Amazing Re-Reviews...

Easyl

"An artist's alternative to the Amiga mouse."

by John Foust

I like to sketch and watercolor. In Deluxe Paint, the mouse was always frustrating. It is difficult to position, as if I were sketching with the point of a brick. I find myself becoming centered on the hot-spot of the mouse, as if the pixels appeared somewhere under the left mouse button.

The Easyl from Anakin Research is an artist's alternative to the Amiga mouse. The Easyl is becoming increasingly popular with Amiga artists, especially with its FCC clearance in the United States early this spring. The new Easyl 1.2 drivers work fine, contrary to the drivers in the previous review by Keith Conforti in this magazine, in the September 1986 issue (volume 1, issue 8) of Amazing Computing.

The Easyl drawing tablet is a wood-framed board about a half-inch thick. The drawing surface is the size of a regular sheet of 8 1/2 by 11 paper. According to the manual, it can discern 1024 by 1024 points.

The tablet is connected to an Amiga expansion box that stands about an inch and a half above my Amiga, and is about an inch and quarter thick. It is connected with a cable to a DB-9 connector on the back of the expansion box. The box does pass the bus, and I have connected a hard disk to the other disk without any trouble. A small red LED is on the box, and it lights when the pad is pressed upon.

Both the Easyl and the mouse can be active at the same time. In other words, you can draw with Easyl, set down your stylus, and use the mouse, if you prefer. The surface of the Easyl makes a good running track for the mouse, even if the tablet is use. The Easyl gives control of the mouse buttons, as well as a more natural drawing interface. The mouse buttons are present as two bubble switches on one side of the Easyl tablet.

To use the Easyl with existing software such as Deluxe Paint or Aegis Images, you must first run a driver. This comes in several flavors, one each for left and right handers, and versions for drawing with or without the bubble buttons pressed. Most people would prefer to draw with the driver that simulates the mouse button being pressed when moving the stylus on the pad.

Easyl comes with a simple drawing program called Easyl. It doesn't offer much over Deluxe Paint or Images, but it works. It has some features specific to the Easyl, such as the ability to use the Easyl tablet in landscape (low rectangle) and portrait (up and down rectangle) drawing modes.

The full source code to the Easyl drivers is also supplied, along with full instructions for setting up assembler and C compiler disks to compile them. The version of the manual I had was oriented to AmigaDOS 1.1, however.

Drawing with Easyl

Even with a sheet of drawing paper over the tablet, the drawing surface is very responsive when pressed more than lightly. An artist can sketch lightly on the paper, without actually drawing on the screen, and then resketch over the drawing to commit the image to the screen.

In Deluxe Paint, the Easyl cannot keep up with rapid drawing strokes. I found the tablet couldn't keep up with handwriting speed of a signature, for example. The documentation claims this is a software-induced speed limit to make the tablet act more like a mouse. The highest capture rate is 250 points per second, according to the manual.

I spoke with Anakin's Jeff Evans at the January Consumer Electronics Show, where Anakin was demonstrating the Easyl in the Commodore booth. Evans was drawn to the Easyl for very specific reasons. "My initial interest in the Easyl was from an animator's and an illustrator's point of view. I was interested in a freehand drawing tool and a tool for doing frame-by-frame animation. For that, it was perfect. It was the only tool that I found for a computer that was was satisfying for those purposes. I've found that a professional designer can transfer a very free, lively rough sketch to the computer with whatever degree of precision they want..."

In animation, each frame of the sequence must be properly aligned with the next frame. Animation drawing paper has two holes at one edge so it aligns precisely on drawing table pegs. Evans fastened a set of these animation register pegs to the top of an Easyl, and hopes to get a patent awarded for that.

Because of changes made in AmigaDOS 1.2, Anakin upgraded the Easyl software. It was sent free to all registered owners. The new software did not include an upgrade to the Easyl drawing program, but only for the standalone drivers that integrate the Easyl into other programs.

I showed the Easyl to a friend who does video production with the Amiga. She has grown accustomed to using the mouse for freehand drawing, and found the Easyl somewhat hard to

continued...

use. Most of this was due to her drawing technique. In art classes, she learned to sketch lightly in case she needed to erase and correct her work. The Easyl was not responsive to her lightly drawn lines. She sometimes used the pencil on its side, to get thicker, soft lines, and the Easyl did not respond to that in the way she liked. If she could have spent more time with the Easyl, I think she would have liked it more.

So, satisfaction with the Easyl might be a matter of personal artistic taste. Learning to draw with a mouse is an artifically acquired taste, but many Amiga artists have grown accustomed to it. I truly enjoyed sketching with Easyl. I had never before used a drawing tablet. In my opinion, Easyl's eye-hand coordination is much more natural than a mouse. To me, drawing with Easyl is much nicer than using a mouse, even if it means an adjustment in drawing pressure.

•AC•

Easyl $499.00

Anakin Research, Inc.
100 Westmore Drive Unit 11C
Rexdale, Ontario
Canada, M9V 5C3
(416) 744-4246

Aegis Animator Scripts and Cel Animation

by John Foust

Aegis Animator can present two types of objects on the screen. The first are polygons. These are closed curves made up of line segments. They have only a single color. The second is windows, also known as brushes, cels or rasters. These are limited to rectangular shapes. They are cut as brushes from a picture created in a painting program, such as Aegis Images or Deluxe Paint. They can be composed of up to 32 colors, including the transparent color, so they don't have to look rectangular. (In this article, I will use "brush" and "polygon" somewhat interchangably.)

I wanted to use Animator to do simple cel animation. "Cel" is short for "celluloid;" it refers to the clear plastic drawing media used by cartoonists. By drawing on a clear sheet and moving that over a background, the whole frame image didn't have to be redrawn for each frame. For example, arms and legs of a character could be drawn in different positions and overlaid on a stock cel of its body.

You have surely seen this type of animation before. Most cartoons on television today use very crude forms of cel animation. Some only change the cels ten times a second. In fact, some of today's cartoons are computer-generated, using the computer to generate the intermediate cels with techniques similar to Animator's "morphing." Older cartoons did not use cel animation as much today's cartoons. Cartoonists of old painted many more backgrounds and character positions by hand than today's cartoonists.

Scripts

For any animation you create in Animator, the action is represented in a script. Animator stores and loads a human-readable text with all the information about an animation - where each object is, how far it moves, the name of each window or brush.

If you wanted, you could create complete animations without using Animator itself by writing scripts. You could edit text files in the format used by Animator, and load these into the program. The program accepts these scripts as if it created them itself using the icon and mouse interface.

The script language isn't difficult to learn. If you ask, Aegis Development will send you a technical reference sheet describing the script language. While this document was of some help to me, it is not complete, and does not cover the majority of script commands. Some people learn the rudiments of the script commands by studying example scripts. An example script is shown in Listing Two.

Cel animation is painstaking work in Animator. My example is a fish. The fish has a body that could stay the same from frame to frame. It has a set of front fins that should change as it moves, flipping from forward motion to backward motion. It has a tail that should move, too.

I first drew the entire fish in Deluxe Paint, without regard to splitting it into separate parts. I made a copy of the fish, just to be safe, then cut out a brush of the area that included the front fins, and placed this front fin brush down on another part of the screen. I used the transparent color to remove the body part of the fin brush, and erased the fins from the body of the fish as well. This process repeated for the back fins.

Then I created the alternate images of each fin, fins flipped in different position. By alternating between these two fin cels, a more pronounced illusion of motion is created. Everything must be saved as brushes with filenames with a '.win' extension, so I saved 'body.win', front fins 'f1.win' and 'f2.win', and back fins 'b1.win' and 'b2.win'.

In addition to the fish, I drew a background picture of a fish tank, complete with gravel, bubbles and weeds, and saved this separately with a '.pic' extension. Animator will not recognize ordinary file names. Its file requesters only recognize files with the proper extensions.

Cel animation

To animate this fish by hand, you would first load the body onto the screen, and position the fins over the body in the correct spots. Change to 'Select Polygons' mode, and click on each brush to select the group. Select 'Path', and move the composite object ahead a small amount. Then select 'Next Tween'. When Animator plays this back, it will move the whole object smoothly from the starting point to the ending point. This process is called 'tweening', and the sequence called a 'tween'.

In the next tween, the fish should have the alternate fin images. By hand, delete the fins using 'Destroy', then load each alternate fin image and place it in the correct spot on the body. As you might guess, it is difficult to position a brush this exactly. (Many advanced Animator users mark a small spot on the body object and align an alternate image brush against that.)

Repeat this process for the entire animation. If you want to flip the fins every ten pixels or so, it takes a long time and a lot of detail work. After creating a few moments of cel anima-

continued...

tion, I knew an easier method was at hand. What are computers for? If Animator scripts are just text files, then a program can generate Animator scripts. To alleviate this extra effort, I wrote a short AmigaBasic program to write portions of Animator scripts for cel animation. It is shown in Listing One.

Automatic scripts

Before I described this program, let's discuss Animator scripts and the commands inside, using the example in Listing Two.

The first few lines of an Animator script are created automatically by the program, the rest represent all the commands you gave to create an animation. Tweens are separated by a single blank line. In Listing Two, the first tween "paragraph" is automatically generated when you save the script. The first line of a saved Animator script looks like this:

```
*script ram:work.script 1 320 200
```

All commands within an Animator script have this general form. The first character is asterisk '*', followed by a command word. This is a 'script' command. The command type is followed by information relevant to that command.

In this case, the 'script' command records information about the entire script. First is the name of this script file, 'RAM:work.script'. I saved this file to the RAM: disk for speed. It turns out Animator ignores this filename, so this script will work from a floppy as well. Next is the number of tweens in this animation. There is certainly more than one tween in this script, but again, this number is ignored. Next is the resolution of the screen, in pixels. Animator works in low resolution, so these numbers are always 320 and 200.

The next line of the script is a 'version' command. This command is presently ignored by Animator.

```
*version 14
```

There have been three versions of Animator, 1.09, 1.10 and the most recent version, 1.20. In general, the Animator script format shows signs of deliberate flexibility, especially in terms of future extensions. There is a lot of extra information in a script that could be used to improve the program in the future. As a programmer, I recognize that future versions of Animator will be able to read scripts created in with older Animator versions.

Next is the 'ground_z' command. It defaults to 512. Numbers greater than 512 means than an object is deeper into the screen, so it appears farther away, and behind all other objects. Objects in Animator can be layered and give the illusion of three dimensions. This sets the depth of the screen plane.

```
*ground_z 512
```

Next is a 'speed' command. This is a default global speed for playing back the animation. We have no need to change this.

```
*speed 20
```

Finally, in this example, there are several 'define' commands. These alert the Animator script interpreter to load these brush files from the disk because they will be needed in this animation. This block of text is the first file generated by the AmigaBasic program.

```
*define AMIGA_BITMAP body.win
*define AMIGA_BITMAP f1.win
*define AMIGA_BITMAP f2.win
*define AMIGA_BITMAP b1.win
*define AMIGA_BITMAP b2.win
```

This AmigaBasic program could have easily generated all the above standard information in the start of a script. For flexibility, it only creates script commands that must be added to a pre-existing scripts. On my work disk, I created a simple script. The only action I performed was loading the background picture of the fish tank. This script only contains two paragraphs, one with this general header information, and one loading the picture and setting the color palette. The two files generated by the AmigaBasic script program are inserted into this basic script using a text editor, such as the AmigaDOS 'ed' editor.

After the 'define's comes the first blank line, marking the next tween. In this case, it is the first tween. Each tween begins with a tween command. Only the second number needs to be set in a script. This is the length of the tween, measured in the units particular to Animator. Two hundred of them take a second. You can leave the others at zero.

```
*tween 0 200 200 8
```

There are a series of commands that all being with the 'act' command. Next follows a number, but this is ignored on loading. Next is a command type, in upper case. There are dozens of 'act' command types, I cannot describe them all here.

```
*act 10 LOAD_BACKGROUND -2 tankback.pic
```

This command loads an IFF picture as a backdrop, behind the animation action. I drew a picture of a fish tank, complete with gravel, bubbles and weeds. The number between 'LOAD_BACKGROUND' and the filename is ignored.

```
*act 101 INIT_COLORS -1 0 32
         (0 32 176) (0 80 240) (32 96 208) (64 112 192)
         ...
```

This is a color map, with number triplets that represent each of the 32 colors available on a low resolution screen. In Animator scripts, the values range from zero to 240. A trio of zeroes means black, and (240 240 240) means white, a color formed by mixing full-scale red, green and blue. Perhaps this maximum white value seems unfamiliar to you. Most Amiga

programs display RGB color values as numbers from zero to fifteen, instead of 240. All Animator color values are multiplied by sixteen.

Why would they do this? If a future Amiga has more than 4096 colors, then that machines' red, green and blue components will vary from zero to some number larger than 15. This allows a finer range of colors "between" the current Amiga colors. Also, Animator may someday be ported to computers with different resolutions and color abilities. It has already been converted to the Atari ST. Actually, Animator was first developed on a Sun One computer with a graphics board that could display 256 colors on the screen at once, out of a palette of 16 million colors.

The first color in the fish tank picture palette is a sea-blue. This is color zero, the transparent color. The color triplet is (0 32 176). It has no red component, a green component of 32 divided by sixteen, or two, and a blue component of 176 divided by sixteen, or eleven.

The INSERT_RASTER command places a brush on the screen at a given location. The first part of this action is the name of the brush, previously registered in the 'define' statements in the script. Next is the pair of numbers that position the brush on the screen, measured as the distance from the upper-left corner.

Finally, it supplies the plane of the object. Objects on the Animator screen can be layered, so that they appear in front of or behind each other. The default plane of the screen is 512, and lower numbers make an object appear behind objects of higher numbers. In the example script, the front fins are at plane 500 and the back fins are at 520, so the front fins appear behind the body.

This INSERT_RASTER example positions the body of the fish on the screen near the right-hand middle of the screen, because the lower-right-hand corner of the Animator screen is position (319, 199). (Each number-pair location marks a "pixel," the computer shorthand for "picture element.")

```
*act 8 INSERT_RASTER 0 body.win 279  88  512
```

At first, I guessed that the position of a brush was measured from the upper-left-most pixel of the brush, but close examination of a script file showed otherwise. The registration point of a brush is in the center. Remember that Deluxe Paint grabs a brush in the center, as well.

This certainly makes it a little harder to count the correct offset from each tag-along cel. The AmigaBasic program needs this offset to calculate the correct position for each tag-along cel as it moves. For example, we need to measure the difference in pixels between the center of the fish body brush to the center of each fin brush. This offset stays constant as the fish moves.

Counting pixels in your favorite paint program is not fun. The easiest method to get the correct offsets is to load the brushes into Animator, and then read the offsets from a saved

script file. Line up the brushes as they should appear in the animation. Load the body first, then the tag-along brushes. Then save the script and examine the text. You can use your favorite editor or the AmigaDOS 'type' command. Look for the INSERT_RASTER commands, and subtract the positions of each tag-along brush from the body brush.

Another note about the differences between brushes in Aegis Images and Deluxe Paint. The IFF standard hunk for brushes includes a storage spot for the location on the screen where the brush was cut from. Deluxe Paint does not use this, while Aegis Images does. If you clip a brush in Aegis Images and load it into Animator, it will appear on the screen at the same spot, ready for positioning for the tween. If the brush is from Deluxe Paint, it will be loaded into the upper left corner, or location (0,0), for positioning.

Back to the example script in Listing Two. After inserting the brushes, I used the 'Select Polygons' and 'Path' menu choices to move all the brushes as a group. This command moves all selected polygons by an equal amount, a negative 8, which translates to a right-to-left motion. A positive offset means a left-to-right motion. The complete set of INSERT_RASTERs and MOVE_POLY commands looks like this:

```
*act 8 INSERT_RASTER 0 body.win 279  88  512
*act 8 INSERT_RASTER 1 frt0.win 276  99  512
```

continued...

```
*act 8 INSERT_RASTER 2 rer0.win 299  93 512
*act 6 MOVE_POLY 1 -8 0 0
*act 6 MOVE_POLY 2 -8 0 0
*act 6 MOVE_POLY 0 -8 0 0
```

The action command for motion in a script is MOVE_POLY. The first number is somewhat complicated to generate. It is an internal number to Animator. This example script only has three polygon brushes on the screen at once. They are numbered from zero to two. The body was loaded first, so it is number zero. The fins are polygons one and two.

MOVE_POLY commands use relative offsets, not absolute coordinates. Note the difference between these two commands. INSERT_RASTER commands are absolute positioning values, while MOVE_POLY are relative motions. In other words, MOVE_POLY commands say "Over the course of this tween, move smoothly to the left by eight pixels." If MOVE_POLY commands were absolute, they would say "Over the course of this tween, move smoothly to coordinate (100,50)." Both methods of movement are equally valid, depending on your perspective. I'll get back to this in a moment.

At the beginning of the next tween, the forward-looking tag-along brushes must be replaced by the corresponding alternate image. The KILL_POLY action command removes a brush from the screen. The only value needed by this command is the number of the polygon. In this example, the first fin is deleted.

```
*act 3 KILL_POLY 1
```

If you are making Animator scripts within Animator, the program will keep track of the number of each polygon. If you are writing Animator scripts by hand in an editor - as many advanced users do - then as the animation becomes more complicated, it grows harder to track the polygon numbers.

Consider this example. If an animation has fifteen polygons, and the tenth is deleted, then number eleven becomes number ten, twelve becomes eleven, and so on. From that point forward in the script, the polygons formerly numbered eleven to fourteen are now ten to thirteen. (Remember, enumeration starts at zero, not one, so the lowest numbered polygon is zero. The highest numbered polygon is fourteen at first and thirteen after the deletion.) This AmigaBasic program can generate scripts for complex animations with multiple polygons, within a few limits.

The difference in positioning style between the INSERT_POLY and MOVE_POLY commands hinders cel animation. After moving the fish a small distance with fins forward, you must delete the fin brushes, replace them with the fins-backward brushes at the correct spot on the body brush, select all the brushes, then move it a small amount. (As they say, rinse, lather, repeat.) This is very tedious. Insertions must be specified absolutely, while motion is always relative, meaning the program does none of the work for you, in cel animation.

(Wouldn't it be nice to have an EXCHANGE_RASTER command as well as an INSERT_RASTER command? The EXCHANGE_RASTER command would not need a pair of specific coordinates for a brush, it would simply swap a brush for a brush on the screen. After all, the program records the current position of all brushes. I've heard a rumor that this command will exist in the next version of Animator.)

Program explanation

The program is very simple, as are all good tools. If you have enough memory, you should be able to run Animator, AmigaBasic, and 'Ed' at the same time, alternating between each program as needed.

The program spends most of its time gathering information from the user. First, it needs the number of tweens. This is roughly the number of seconds the sequence should take.

Next, the number of pre-existing polygons on the screen. There may be other polygons on the screen before this object appears. This program addresses the complexities of polygon enumeration in a slick way. By killing a brush, then immediately replacing it, the number of that polygon does not change. Of course, this means that there can be no higher-numbered polygons in the animation. This program can only generate scripts for the last object placed on the screen.

After this, it asks for the filename and plane of the body object, and its starting and ending location on the screen. Then, after asking for the number of tag-along objects and their the filenames, it gets the offsets for each. Each tag-along can have its own plane, so the tag-along fins (or arms or legs or tentacles) could appear to move behind the body object, too.

Finally, it asks for an output script name. This program does not generate the header information for a script, it only generates parts of a script. You must do a little cutting and pasting in a text editor. Two files are generated, one with the name supplied, and one with the same name but a '.def' extension.

These two files must be added to a base script. To make such a base script, start a new script in Animator and load in just the background picture. Save this script.

The '.def' file contains the 'define' commands for the script. This file must be inserted after the 'speed' command in the beginning of the base script. In the 'Ed' editor, load the base script, position the cursor to this spot, and hit 'ESC', and enter 'if/basescriptname.def'. This is the 'insert file' command in 'Ed', it will load a series of 'define's generated by the program. Delete any blank lines between the 'speed' line and the 'define's with CTRL/B. One blank line must follow the 'define's, as in Listing Two.

Now insert the animation script with the filename you supplied. This should follow the INIT_COLORS data, with no intervening blank lines. This should be the last line of the file.

Exit and save with 'ESC' then 'x'.

Move back to the Animator screen, and load this edited base script. It should work like any other Animator script, but relish all the effort it saved.

This program, the scripts and my fish image can be found on one of the new AMICUS disks. This program can be easily expanded. It could be modified to handle more than one alternate image per tag-along with little effort. It would be as easy to let each tag-along images have different planes. Moving the object in a curve instead of a straight line would be a little more complicated, but not beyond most AmigaBasic programmers. Moving more than one composite object at once would be more work. In the future, I hope someone develops a program to create and edit cel animation scripts. For this task, the editor in Aegis Animator is very primitive.

Listing One

```
' Aegis Animator Script Generator
' By John Foust for Amazing Computing

DIM TagName$(10, 1), TagXOffset%(10), TagYOffset%(10),
TagPlane%(10)
DIM TagXPos%(10), TagYPos%(10)

INPUT "Number of tweens: ", NumTweens%

INPUT "Number of pre-existing polygons on the screen: ",
BasePoly%

INPUT "Name of base polygon: ", TagName$(0,1)
INPUT "Plane of base polygon: ", TagPlane%(0)
INPUT "Starting X, Y position of base polygon: ";
BaseXStart%, BaseYStart%
INPUT "Ending X, Y position of base polygon: "; BaseXEnd%,
BaseYEnd%

INPUT "Number of tag-along polygons: ", NumTags%

FOR Tag% = 1 TO NumTags%
  PRINT "Name of tag-along"; Tag%; ": ";
  INPUT TagName$(Tag%, 1)
  PRINT "Name of alternate"; Tag%; ": ";
  INPUT TagName$(Tag%, 0)
  PRINT "X, Y offset of "; TagName$(Tag%, 1);
  PRINT " from "; TagName$(0,1); ": ";
  INPUT TagXOffset%(Tag%), TagYOffset%(Tag%)
  PRINT "Plane of tag-along"; Tag%; ": ";
  INPUT TagPlane%(Tag%)
NEXT Tag%

INPUT "Output script file name: "; OutFile$

OPEN OutFile$ FOR OUTPUT AS #1

' Make a file with .def extension for *defines
OPEN OutFile$+".def" FOR OUTPUT AS #2

' Define the objects in this script
' First the base object
PRINT #2, "*define AMIGA_BITMAP "; TagName$(0,1)

' Then the alternate objects
FOR Tag% = 1 TO NumTags%
  PRINT #2, "*define AMIGA_BITMAP "; TagName$(Tag%,1)
  PRINT #2, "*define AMIGA_BITMAP "; TagName$(Tag%,0)
NEXT Tag%

CLOSE #2

' Calculate the step size in the X and Y direction for each
tween
TweenXStep = (BaseXEnd% - BaseXStart%) / NumTweens%
TweenYStep = (BaseYEnd% - BaseYStart%) / NumTweens%

' Insert the object at the starting point

PRINT #1, "*act 8 INSERT_RASTER "; BasePoly%; TagName$(0,1);
PRINT #1, BaseXStart%; BaseYStart%; TagPlane%(0)

FOR Tag% = 1 TO NumTags%

  PRINT #1, "*act 8 INSERT_RASTER "; BasePoly% + Tag%;
  PRINT #1, TagName$(Tag%, 1);
  PRINT #1, BaseXStart% + TagXOffset%(Tag%);
  PRINT #1, BaseYStart% + TagYOffset%(Tag%); TagPlane%(Tag%)

NEXT Tag%

' Start the object moving for the first tween

PRINT #1, "*act 6 MOVE_POLY "; BasePoly%;
PRINT #1, INT(TweenXStep); INT(TweenYStep); 0

FOR Tag% = 1 TO NumTags%

  PRINT #1, "*act 6 MOVE_POLY "; BasePoly% + Tag%;
  PRINT #1, INT(TweenXStep); INT(TweenYStep); 0

NEXT Tag%

PRINT #1, ""
```

continued...

```
' Update our position

CurrentXPos% = BaseXStart% + INT(TweenXStep)
CurrentYPos% = BaseYStart% + INT(TweenYStep)

' In each tween, remove each tag-along polygon
' and replace it with its alternate, moving the
' whole object as well.  The first tween is done.

FOR Tween% = 2 TO NumTweens%
  PRINT #1, "*tween 0 200 0 0"
  ' Replace each tag-along polygon with its alternate
  FOR Tag% = 1 TO NumTags%
    PRINT #1, "*act 3 KILL_POLY "; BasePoly% + Tag%
    PRINT #1, "*act 8 INSERT_RASTER "; BasePoly% + Tag%;
    PRINT #1, TagName$(Tag%, Tween% MOD 2);
    PRINT #1, CurrentXPos% + TagXOffset%(Tag%);
    PRINT #1, CurrentYPos% + TagYOffset%(Tag%);
    PRINT #1, TagPlane%(Tag%)
  NEXT Tag%

  ' Now move the whole object
  ' Update our position
  CurrentXPos% = CurrentXPos% + INT(TweenXStep)
  CurrentYPos% = CurrentYPos% + INT(TweenYStep)
  PRINT #1, "*act 6 MOVE_POLY "; BasePoly%;
  PRINT #1, INT(TweenXStep); INT(TweenYStep); 0
  FOR Tag% = 1 TO NumTags%
    PRINT #1, "*act 6 MOVE_POLY "; BasePoly% + Tag%;
    PRINT #1, INT(TweenXStep); INT(TweenYStep); 0
  NEXT Tag%

  ' Print the blank line between tweens
  PRINT #1, ""
NEXT Tween%
CLOSE #1
END
```

Listing Two

```
*script ram:work.script 1 320 200
*version 14
*ground_z 512
*speed 20
*define AMIGA_BITMAP body.win
*define AMIGA_BITMAP f1.win
*define AMIGA_BITMAP f2.win
*define AMIGA_BITMAP b1.win
*define AMIGA_BITMAP b2.win

*tween 0 200 200 2
*act 8 LOAD_BACKGROUND -2 tank.pic
*act 101 INIT_COLORS -1 0 32
   (0 32 176 ) (0 80 240 ) (32 96 208 ) (64 112 192 )
   (208 128 0 ) (240 224 0 ) (128 240 0 ) (0 128 0 )
   (0 176 96 ) (0 208 208 ) (0 160 240 ) (0 112 192 )
   (0 0 240 ) (112 0 240 ) (192 0 224 ) (192 0 128 )
   (96 32 0 ) (224 80 32 ) (160 80 32 ) (240 192 160 )
   (48 48 48 ) (64 64 64 ) (80 80 80 ) (96 96 96 )
   (112 112 112 ) (128 128 128 ) (144 144 144 ) (160 160 160 )
   (192 192 192 ) (208 208 208 ) (224 224 224 ) (240 240 240 )
*act 8 INSERT_RASTER  0 body.win 250  100  512
*act 8 INSERT_RASTER  1 f1.win 245  110  500
*act 8 INSERT_RASTER  2 b1.win 260  110  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f2.win 225  107  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b2.win 240  107  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f1.win 205  104  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b1.win 220  104  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f2.win 185  101  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b2.win 200  101  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f1.win 165  98  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b1.win 180  98  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f2.win 145  95  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b2.win 160  95  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f1.win 125  92  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b1.win 140  92  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f2.win 105  89  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b2.win 120  89  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f1.win 85  86  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b1.win 100  86  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0

*tween 0 200 0 0
*act 3 KILL_POLY  1
*act 8 INSERT_RASTER  1 f2.win 65  83  500
*act 3 KILL_POLY  2
*act 8 INSERT_RASTER  2 b2.win 80  83  520
*act 6 MOVE_POLY  0 -20 -3  0
*act 6 MOVE_POLY  1 -20 -3  0
*act 6 MOVE_POLY  2 -20 -3  0
```

•AC•

Quality Video from a Quality Computer

by Oran Sands III

A Discussion of Video & Modifications to the Amiga Composite Video Output

All right, what do your hate the most about your Amiga? The complaint I hear most often is that the composite video output of the Amiga is no good. Curiously though, I never hear anyone exactly state what the problem is. After using Amy for a over a year at my production studio, I think I can offer some help in not only defining the problems, but also solving them.

Problem #1 *- "It's all in your head"*
The first problem is purely psychological. After spending hours looking at a high quality, RGB analog monitor display, seeing your images displayed on a composite video monitor is a great letdown. The two images will never compare well. If you're designing an image for television or tape, check what it looks like often on your composite monitor or on your Amiga monitor itself, using the composite input. Simply switching the source on the front can save a lot of frustration later. The picture won't be as clear and it won't be as vivid. Don't worry about it though, your audience won't be making the same comparision as you. They'll love it.

Problem #2
"Real problems-inherent in composite video"
First, a short note on the different forms that a video picture from a computer may undergo in getting to a display device (see drawing). The computer data of the picture starts life as data in a bitmap. These digital values are converted to TTL level signals representing the Red, Green and Blue (four bits for each on the Amiga, one bit each for the IBM with an additional bit for intensity).

These signals can be converted to Red, Green and Blue analog non-composite video signals. Non-composite means that there is no sync signal combined with the video signal. A separate synchronization signal must be used to display a picture at this point. Composite video is then made by encoding the several video signals together with sync signals to coordinate the vertical and horizontal sweeps of the display scans. The composite video signal can then be modulated by a transmitter and broadcast or cablecast by a RF modulator (RF stands for Radio Frequency).

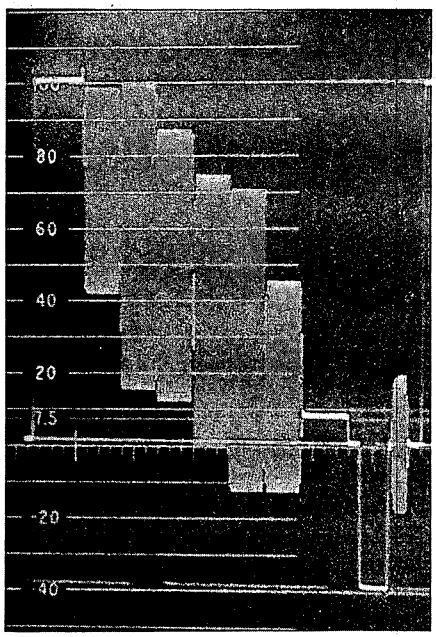
Now the signal has to be demodulated back to a composite video signal which, in turn, must be decoded back into the separate video and sync signals. The signals can then directly drive the CRT of your monitor/tv. Most of our problems are caused by the various conversions of the signal to and from. Every conversion process in video is made up of mostly analog. The signal's resolution and colorimetry are flawed at each point in the process. The best way to avoid this degradation is to eliminate the number of steps the signal must pass through getting to your screen.

The RGB analog display of the Amiga is great because it avoids most of the steps. Although composite video is yet another step along the way, it is the signal of choice for video signal distribution around the world. The system of television we use is known as NTSC RS-170A* and the composite signal must adhere to this standard. To quote Motorola "...NTSC encoding, no matter how rigorously executed, will cause some degree of picture degradation." Since both the NTSC encoding and decoding processes tend to muck up the picture, it behooves us to not do some things that won't survive the trip. For instance, stay away from single or double-pixel thick lines. Not only will they cause flicker but they may be lost altogether in a blur.

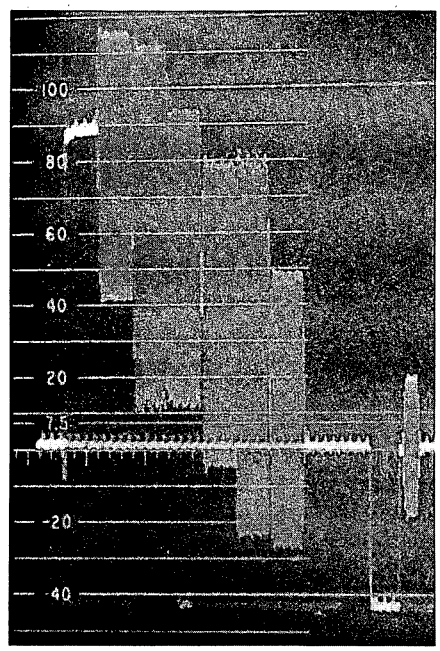
Many fonts are also too thin to be seen clearly once viewed on composite video monitors. A good font should be filled out well. Zuma Fonts are nice in the larger sizes. Very subtle shading will blend together, which can often be a boon, but at other times, can be a curse. Checking your work on a composite monitor can be great security (still got that old 1701 sitting around?) . Remember, watch your contrasts in color and brightness; look out for too-thin lines or dots and use thicker fonts.

Problem #3 *- "This one's for real, but fixable."*
This problem was discovered when it was brought to my attention that you couldn't make proper color bars with a paint program. The color bar signal is a standard in the video industry and the bars of color are the result of mixing the red, green, and blue signals (all at full intensity) in different combinations. The order of the vertical bars (and they MUST be vertical) are as follows:

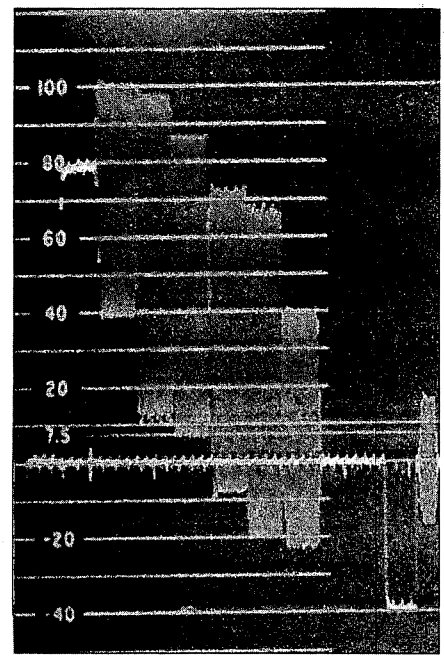
continued...



1A - EIA Color Bars



2A - Uncorrected Amiga Signal



3A - Corrected Amiga Signal

COLOR	RED	GREEN	BLUE
white	on	on	on
yellow	on	on	off
cyan	off	on	on
green	off	on	off
magenta	on	off	on
red	on	off	off
blue	off	off	on
black	off	off	off

"On" means a value of 15; "off" is a value of 00. The colorbar picture on the Genlock demo disk is sufficient, but it would be better to make your own in hi-res (and in overscan mode, if possible).

This signal results in a certain type of composite video output which should have certain electrical properties. The failure of the Amiga to react correctly to such an image clued me into the existence of what we call an encoding error.

The encoding process uses the three RGB signals and, by adding them via a matrix method, results in the composite video signal. Should the signals not be added correctly, the composite signal won't look the same as the RGB picture. The very hues and color intensities themselves will not be right. By using television instruments known as waveform monitors and vectorscopes, this problem was quickly documented.

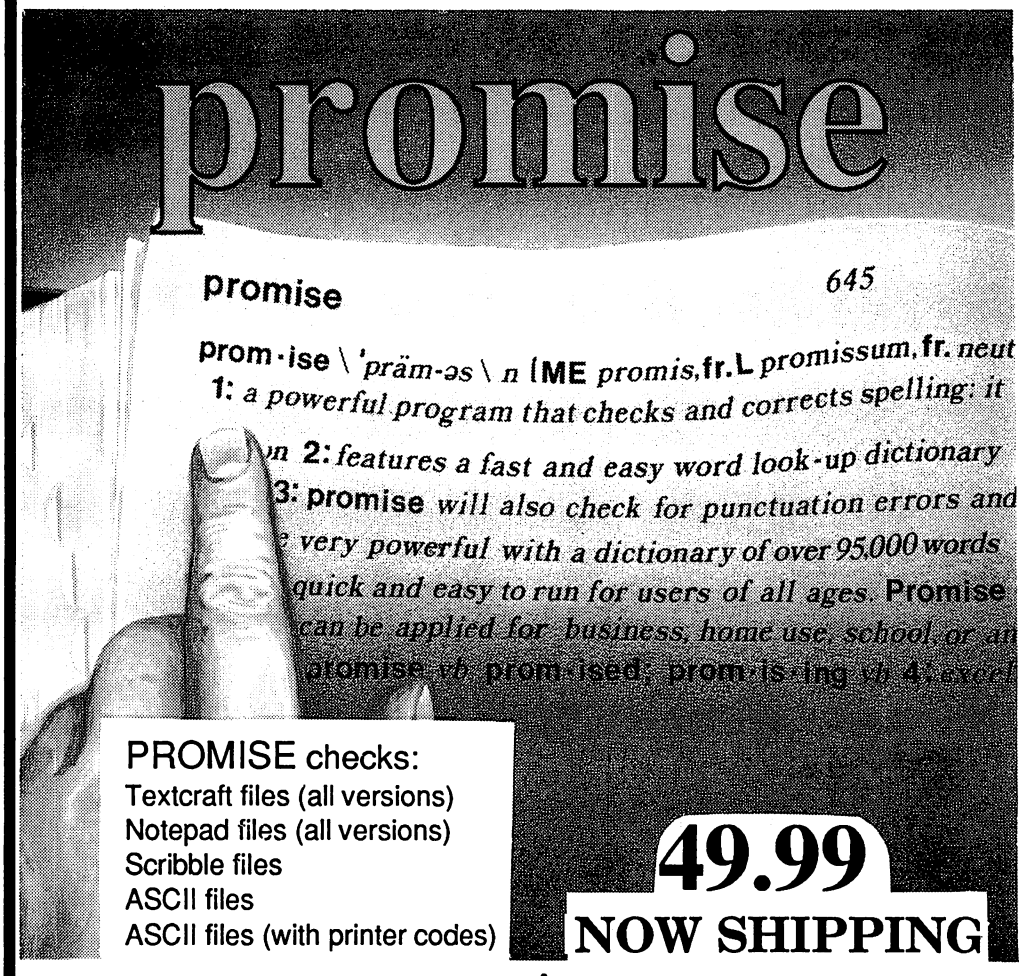
Pictures 1a & b show the correct readings for a color bar signal generated by a Textronix 1470 signal generator. Picture 1a is a video signal shown on a waveform monitor allows measurement of the brightness and saturation. Picture 1b is the same signal shown on a vectorscope and measures saturation (distance from center of circle) and hue (angle from baseline). Note that no parts of the 1a signal exceed 100 IRE units (luminance levels) or -40 IRE. The hues of 1b are measured in terms of degrees and you'll notice that they all fall within marked boxes only two degrees wide (within boxes five(5) degrees wide). These 2 degree boxes are considered to be the acceptable margin of error for a quality signal (it is generally accepted that the human eye can't detect less than seven degrees change, but I'd contest that).

Now, let's measure the standard Amiga. Picture 2a shows that the signal is too bright in the wrong places and generally dim in others. The saturation levels are wrong as well. The colors don't even land within the five degree boxes (except for yellow and blue). Obviously an encoder problem. After obtaining a set of schematics from Cardinal Systems, the culprit appeared to be the Motorola MC1377P chip. This device does all the encoding process within itself. After obtaining the data sheets on this chip, I found that Amiga implemented the design according to Motorola's instructions. Except for one little resistor.

"Enter the Villain"

The encoding process references two axis: the R-Y axis and the B-Y axis. The R and B are the red and blue signals and Y is the luminance signal (derived from the RGB signals themselves). The actual relationships are as follows:

Y = .59G + .30R + .11B
R-Y = .70R - .59G + .11B
B-Y = .89B - .59G + .30R



Special

Introductry Price

Complete Communications Package

300/1200 1 Year warranty

300/1200 Fully Hayes compatable Modem - 2 Year warranty

$129.00
(Modem, Cable & Software)

300/1200/2400 Fully Hayes compatible modem CCITT - 2 Year warranty

$249.00
(Modem, Cable & Software)

Introducing promise

At last, the introduction of a new generation of SPELL CHECKING PROGRAMS. Promise contains many features to meet the expectations of demanding users.

Features:

- A 95,000 word dictionary that resides in memory and uses advanced indexing methods.
- A spell checking rate of 18 words per second! (1080 words per minute).
- ***PROMISE*** features a SPELL HELP that actually helps you spell any word correctly. SPELL HELP will operate with virtually any word processor.
- SPELL HELP takes full advantage of the Amiga's multi-tasking capabilities, allowing the spell help feature to be used with most word processors.
- The ability to create custom dictionaries at the mere click of the mouse button.
- Total mouse and gadget control, giving an ease of use never before available in spell checking programs.
- ***PROMISE*** will check for rudimentary punctuation errors.

Promise makes word processing faster and easier. *Promise* is a must for any serious writer, student, or business person.

Amiga is a trademark of Commodore Amiga

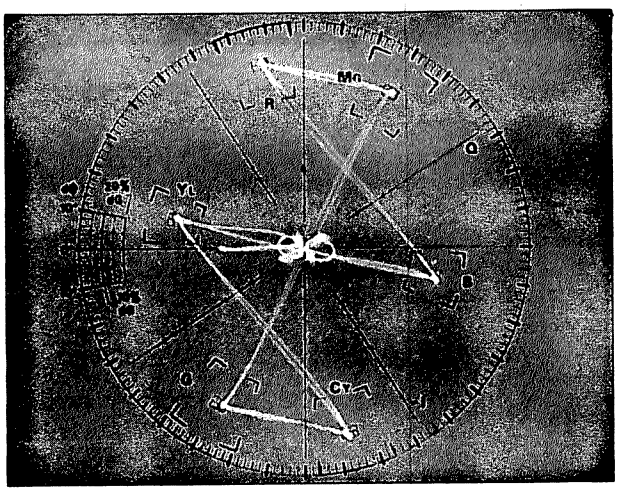
Requires 512K Amiga.

THE OTHER GUYS

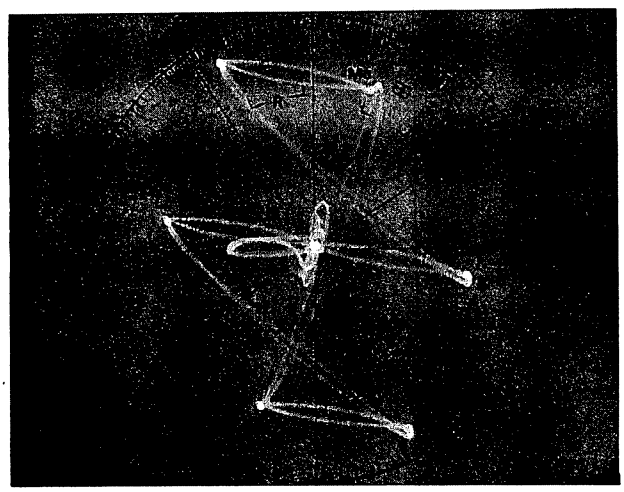
(800) 942-9402

The Other Guys, Amiga
55 N. Main Suite 301D
Logan, Utah 84321

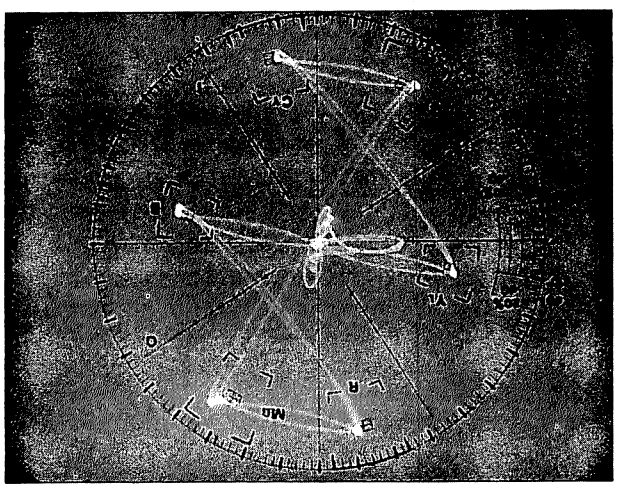
VISA MasterCard



1B - EIA Color Bars



2B - Uncorrected Amiga Signal



3B - Corrected Amiga Signal

Looking at the vectorscope pictures, you'll notice a vertical and horizontal line intersecting at the center of the circle. These lines represent the R-Y (vertical) and the B-Y (horizontal) axis. The encoder should maintain these axes at a 90 degree angle, but there is an adjustment for altering that angle. Changing that relationship readjusts the percentages of each color in the above equations. Amiga chose to use a resistor value that changes that relationship to about 97 degrees. Don't ask why. I don't know. Motorola indicates in their literature that using no resistor at all should provide correct alignment, so why Amiga decided to shift things by 7 or so degrees is beyond me.

Here's the good news. Simply replacing the resistor will allow your Amiga to fall back to not only within the five degree boxes, but almost perfectly to the two degree areas. If you would plot a line through the points representing each color before, and then after, the correction you'd get parallel lines all aligned with the horizontal axis (B-Y). In fact, all adjustments made using different resistors will simply move the plot points along these lines (it's possible to move the points to both sides of the correct spots).

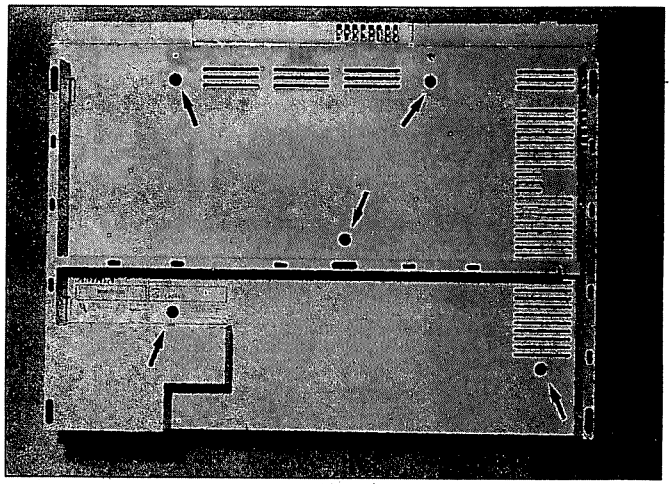
Now that we know what to do, let's get to it. But first a word from your sponsor.

DISCLAIMER

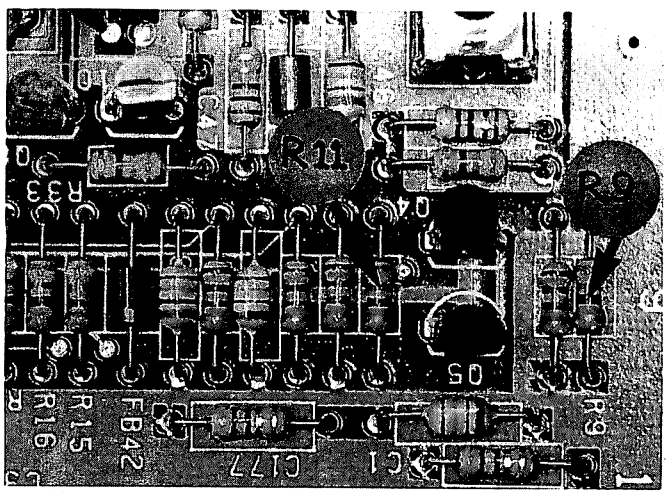
Opening the Amiga and/or making any modifications internally will void your warranty. This magazine, its authors, editors and myself cannot be held responsible for any damages incurred as a result of attempting this modification. Do not attempt this yourself, unless you are familiar with the techniques of soldering on integrated circuits and heat- and static-sensitive devices.

Okay, with that out of the way, turn off the power, disconnect all periphials and open the Amiga case by removing the 5 phillips-head screws on the bottom. Now, remove the tin shield by undoing all the screws (don't miss those holding the shield around the ports) and straightening the metal tabs with a pair of needle-nose pliers. Once the cover is off, look at the rear corner near the power supply (close to the composite video output). You should find the MC1377P chip. Now familiarize yourself with figure 1 and the parts that surround the chip. Look for resistor R140 (marked on the board; it has yellow and purple stripes). If your Amiga was bought prior to June 1986, replace R140 with a 470K ohm 1/4 watt resistor (Radio Shack #271-1354). If yours is a later model, then simply eliminating R140 will correct the problem. If you don't want to desolder R140, then merely clip the lead of the resistor that is nearest the chip and bend it out of harm's way to the side. I wish I could give you a better idea which machines are which, but there is definitely a difference between the early machines and the newer ones. Regardless, your machine will be within 3 degrees of its optimum setting in the worst case. Evidently, Motorola is getting better at making these chips as time goes by. . These repairs will almost perfectly align the color hues and saturations (see photos). Don't attempt this unless you know what you're doing! You could easily fry the chip which is only $3.50 and have someone else replace it for $60.00 or more.

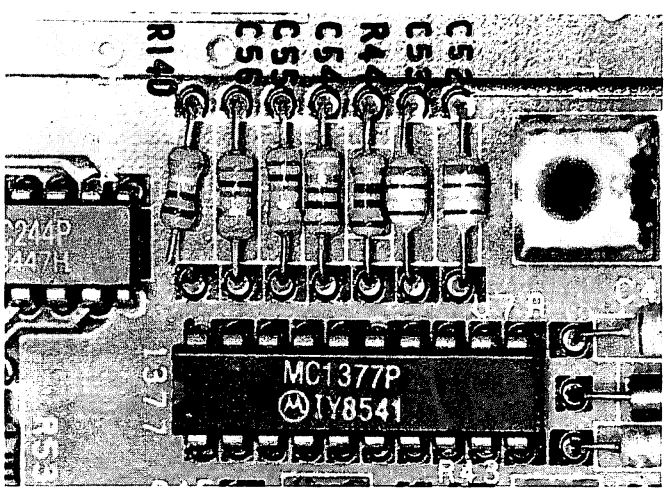
continue...



Location of Screws on Amiga™



Location of R9 and R11



Location of Resistor R140

Now the composite video output should be pretty close to what you see on the RGB monitor. Not as vivid, not as clear perhaps, but at least red is still red (not purple) and green's green. Be aware of the fact that many pieces of video equipment will attenuate the signal automatically if too high (and Amy's output is still high at this point). Sometimes this is good, but not always. If you're transmitting the signal, such extra brightness could distort the image. If you're only displaying to a monitor however, the extra brightness will give you a more vivid picture. If this is a bother, let's fix it also.

In order to reduce the amount of video, we will have to replace a resistor. The 75 ohm resistor (R9), attached to Q4 in combination with the 75 ohm impedance of your VCR or monitor's input, makes up a voltage divider. This arrangement allows for maximum power transfer, but unfortunately in the Amiga's case, too much video as well. Since we can't tamper with your VCR or monitor, we'll change the Amiga's resistor. Replacing R9 means a little soldering, so remember our caveat stated above. Check out the photo to find R9 (the one with purple and green stripes). The best value resistor to use is 100 ohms (1/4 watt). You'll find it for sale at the local Radio Shack (part# 271-1311). You'll need to decide if this is something you really need. Sidenote: The RF jack also has a composite video output. This output gets its signal from the same chip, so our R140 mod affects it as well. The R9 mod doesn't, however. The R9 mod can be performed on R11 (also shown in the photo) to make the same change on the RF jack. Personally I'd modify one and leave the other alone (one for the VCR and one for your monitor).

This modification has been made on several Amigas and all have responded the same. I have a hunch that the early machines might have had a problem that the R140 would have cured. The only problem with this modification is that there is no way to tell if it's made a positive change, without using a waveform monitor and vectorscope. So, at his point let me say "trust me". The Amiga will never put out a "broadcast quality" (a phrase ranking right up there with "user-friendliness") signal on its own, regardless of the amount of modification. But at least an industrial quality signal is distinctly possible. The only noticeable differences in the Color Bar signal waveform and that of the Amiga's is caused by the Amiga's lack of a 7.5 IRE unit offset at the baseline (this is known as the setup or pedestal). This is a hangover from days when TV's couldn't handle dark blacks (defined as 0 IRE units), so black was defined as dark gray (7.5). Today, the setup is not really necessary and leaving it out allows you to acheive a greater range of contrast.

If you have a Genlock 1300, you should be aware of the fact that it uses the same MC1377P to create the video output. This time Amiga used a resistor with a value of 1.5 Megohms. This value is about the same as infinity to the chip, and that's the same as no resistor at all! So, it looks like the genlock designers caught on. Now, about that hue control on the back. Bad news, it alters the hues in much the same way our resistor R140 did inside the Amiga. A true tint control simply causes the signal on a vectorscope to rotate about the center of the circle, with all phase relationships being maintained (i.e. red is always 244 degrees from blue). Don't be surprised if you find yourself able to adjust the picture so all colors are

correct, except for one. When in doubt as to when the hue is set as close as it'll get, check out the flesh tones. Whether any other color is correct, everyone expects flesh tones to look correct. I don't have a fix for this yet, but maybe someday.

I hope you find this modification as useful as it is simple. Video on the Amiga is a wonderful thing and sometimes quite awe-inspiring. Letting the color come through can only help!

P.S. Another magazine has started a rumor that the cable for a C-64, when plugged into the RF jack, gives better video. Poppycock! This is probably based on the fact the cable (8 pin DIN) furnished separate luminance and chroma signals to the 1701 monitor. Frankly, the cable had nothing to do with it! Those signals appear at the C-64 video port because Commodore designed it that way. The cable just gets it to the monitor. Those signals are what will become encoded into the composite output of the C-64 (a midway step between the RGB signals and the composite video). Trying to decode the composite video of any signal to arrive at separate luma and chroma signals will result in a poorer signal than if you had those signals before the encoding process. Noise and degradation occur during encoding and decoding. Besides, why bother? Your monitor does that step anyway, enroute to making a picture from the composite video.

P.P.S. It has been mentioned several times in this and other publications that the Amiga's output (or the Genlocks) must be run through a time base corrector before it can be used. There is a major misconception here. The Amiga's output is perfectly stable and needs no correction of its timebase (stability of its clock). The Genlock's output is as stable as what you feed it. However, most timebase correctors also have built-in a processing amplifier, which allows for individual adjustment of the sync level (that negative-going pulse in photo 1A) , video gain, color intensity (chroma) and hue (tint). It may be that you would like to control these items. If so, buy a processing amplifier (known as a proc amp), not a timebase corrector. The cost can be $500 vs. $6000. As for needing to adjust these parameters, hopefully the mods described in the article make a moot point of needing a proc amp.

* NTSC National Television Standards Committee

•AC•

Amiga schematics
Cardinal Systems
14840 Build America Drive
Woodbridge, VA 22191
$24.95

Application notes for the MC1377P and literature #AN932
Motorola Technical Information Center
Motorola Semiconductor Products, Inc.
P.O. Box 20912
Phoenix , AZ 85036
Free

Is IFF Really A Standard?

by John Foust

"IFF" is one of the most common terms you hear in the Amiga software world. IFF stands for Interchange File Format. It describes a standard way of storing data in files on computers. It is not Amiga-specific. IFF is a general standard for data interchange. It is not specific to graphic data, although that has been its most popular incarnation on the Amiga. IFF encompasses animation, text, sound and music.

The IFF standard was drafted in the early days of the Amiga by a committee of Amiga developers, mostly people from Electronic Arts and Commodore-Amiga. They hoped to spread this standard to other computers.

Although the Amiga has a standard file format, it does not mean Babel has been avoided. By and large, IFF pictures can be used in any Amiga graphic program. However, the popularity of IFF has been countered by the inflexibility of some programs that use it. It can be a confusing quest to move an IFF image from one program to another, without distortion in shape or color. This article discusses the reality of data interchange on the Amiga. It should serve as a road map for moving graphics data between Amiga programs.

Amiga resolutions

First, a little background on Amiga video. The Amiga has four graphic resolutions. "Resolution" describes how many colored dots can be drawn on the screen, measured across and up-and-down. Many people use the term "pixels," instead of "dots." The four resolutions are 320 by 200 pixels, commonly called "low resolution;" 320 by 400 pixels, sometimes called "interlace" or "low resolution interlace;" 640 by 200 pixels, called "medium resolution;" and 640 by 400, called "high resolution."

The Workbench is medium resolution, for example. As you might see, there are two independantly selectable horizontal resolutions, plus the option of interlace which doubles the default vertical resolution of 200 pixels. High resolution mode is interlaced, as well. All interlaced modes give the flicker that occurs when bands of contrasting colors are displayed.

Again, computer people have a tendency to shorten and idiomize terms, so you might hear the word "resolution" pronounced as "rez" and spelled as "res," as in "low res" and "high res."

All graphic images take up a certain amount of memory space ,whether the picture is on the screen or on a disk. The number of colors used in an image also affects storage size, so there is a relation between resolution, color, file size and memory consumption.

Within each resolution, only a certain number of colors can be on-screen at once. In low resolution, up to 32 colors can be displayed. The number of colors possible on a given screen is always a multiple of 2, so low res screens can be limited to 4, 8 or 16 colors if desired. Medium and high resolutions can display a maximum of sixteen colors, or 8, 4 or 2 if desired. HAM can display 2 to 4096 colors on the screen.

HAM

The Amiga has another unique graphic mode possible called HAM. This is short for "hold and modify." This is the famous 4096 color mode. HAM is only possible in the two low resolution modes. HAM has limitations as well. It cannot represent all sharp color transitions. It takes, at most, three pixels to go from one color to another on a single horizontal scan line. A programming technique can improve this, so that more transitions happen in less than three pixels. But this cannot eliminate the color streaks that typify a HAM color transition.

continued...

It is not well-known that other programming tricks can increase the number of colors possible in non-HAM modes. The color palette can be changed on the fly, as a screen is being displayed. As many as eight of the 32 colors in a low resolution palette can be changed on a single video scan line, so the entire palette can be changed in four scan lines.

This is what happens when you slide a screen down to reveal the screen below. The Workbench reloads the color palette for the lower screen, in the short time it takes to display the four scan lines that form the border on the screen being lowered. This technique has limited utility in most applications, but it may surface in future games. Some screens in the game "Pawn" use this to get more than 32 colors on the screen at once.

Another video mode is possible in some custom video chips. It is not present in most Amigas. These chips were produced as an experiment and only a small number of Amigas have this chip. This mode is called "half-bright" because it doubles the number of colors in a low resolution screen by optionally drawing pixels with half the brightness of an existing palette color, creating twice as many colors by adding as many shades. No programs currently use this mode, but future Aegis programs are planned to accomodate Amigas with the half-bright graphics chip.

The Amiga has another ability that makes it stand out against the crowd of computers with nice graphics. It can render graphics into the borders of the screen, so that the entire video field contains computer-generated graphics. This is called overscan. Most computers generate a fixed border around their image and this part of the screen can never be changed.

IFF principles

An IFF file is characterized by hunks. A hunk is a short identifying tag present in the data. All hunks have a similar format that can be decoded by any program that professes to be "IFF compatible."

Each hunk has a human-readable name of four characters. You might hear of a 'CAMG' hunk, a 'BMHD' hunk, a 'CRNG' hunk. (You can see these hunks in an IFF file if you use the 'type opt h' command to view the file in hexadecimal.) Hunk types are defined to implement common file formats, such as IFF pictures, sound samples and text. A single IFF file will have several kinds of IFF hunks. In the file, each hunk is followed by its data. Another hunk may follow after that point.

Why might one program reject an IFF file? If a program has created an incorrect IFF hunk or added improper data to the hunks, it might not be considered as IFF by other programs, even if it can be re-loaded by the same program.

Rejecting an improper IFF file is the proper response for a program. The IFF specification goes further than this, however. To maintain IFF compatibility in future programs, programs written now should ignore unrecognizable hunk types. In this way, new IFF hunks can be defined in the future and your old copy of Deluxe Paint will still be able to read the newer IFF files. Also, when asked to load a color palette, you could specify any IFF file that had color information in it (whether it be a brush, picture or only color information.) Most programs are not this flexible.

To be this flexible, a programmer must create an IFF reader that is more complex than might be expected. With a static, unchanging file specification, a program can expect to find valuable data at specific locations in a file. This sort of program might be called "quick and dirty" because it might load IFF pictures faster than a more complex, but flexible reader. It is not guaranteed to work in the future. Often, Amiga programs are somewhere between the two extremes: they may be tolerant of unknown hunks, but do not load all information they possibly could.

Nomenclature

Some paint programs use different terms to describe the same thing. To move data between programs, you must first define common terms used to describe objects.

For example, Graphicraft 1.1 has a "Save Brushes" option. Deluxe Paint can mark an area of an image and save it as a brush. Graphicraft does not save brushes in the Deluxe Paint sense. Instead, Graphicraft allows the creation of custom

paint brushes beyond the standard blocks, bar, circles and dots. Its "Save Brushes" option saves these custom paint brushes. Surprisingly, Deluxe Paint II can load the Graphicraft brush set, but it loads the sixteen custom brushes as a single brush.

What Deluxe Paint calls brushes, Images calls windows. Aegis Images windows also include the location on the screen where the brush came from, while Deluxe Paint brushes do not record this location. Aegis Animator uses this brush location, while Deluxe Video does not.

The only way to move color palettes between pictures in Deluxe Paint is to use a brush because the color palette is stored with the brush. Aegis Images can save a color palette separately, but these files are not recognized as brushes by Deluxe Paint. The only way to move an Images color palette to Deluxe Paint is with a brush, but there is even a complication to that.

Aegis Images has two versions, Images and Images-HR. Images is 320 by 200 resolution, called low res in Deluxe Paint. Images-HR stands for "high resolution," but it is only 640 by 200 resolution. Deluxe Paint calls this medium resolution.

Converting resolutions

Converting pictures from one resolution to another is not easy. Graphicraft only works in low resolution. It loads pictures very slowly, but it does its best to convert medium, interlace and high res pictures to low res. The original Deluxe Paint can only load pictures in the currently selected resolution. The only way to change resolutions was to restart the program from the CLI. There was no way to change resolutions from the Workbench. Deluxe Paint II will load pictures of any resolution into the current resolution, but it does not strictly convert pictures to other resolutions.

The Butcher program from Eagle Tree Software will convert pictures from one resolution to another, as well as converting HAM pictures to low res 32 color. It has very handy features for manipulating color palettes, sorting, remapping and reducing the number of colors used.

Butcher does not do as well as the Digi-View software for converting HAM to low res. The Digi-View software has powerful image manipulation abilities. It can convert HAM to any other resolution with very good results, using proprietary dithering algorithms. It can convert between any resolution and remap color palettes. It serves as a crossroads for many people who need to convert images. Unfortunately, the program is not sold separately from the Digi-View digitizer.

There are two kinds of color cycling hunks in use. Graphicraft uses one called 'CCRT.' This is quite flexible in terms of color ranges and very adjustable timing for color cycling. The Deluxe Paint II color cycling hunk is not as accurate in controlling timings as the Graphicraft hunk. In particular, the Deluxe Paint II hunk does not work well on European PAL systems.

However, the Graphicraft cycling hunk is not in widespread use because Deluxe Paint II uses a different hunk type to set color cycling, called 'CRNG.' Deluxe Paint II brushes are saved with color cycling information, while original Deluxe Paint brushes are not. Deluxe Paint II also incorporates cycling direction in its CRNG hunks. Deluxe Paint cannot set the direction of a color cycle.

General problems

Some problems are specific to particular Amiga programs, but still deserve to be highlighted in a summary of graphic program problems. In Deluxe Paint II, if you are using the HSV sliders to adjust colors, the sliders are not numbered, so if you are using the HSV sliders to adjust colors, you can't reproduce settings accurately.

Aegis Images does not change the pallete when menus are selected, so if you have set certain colors to very dark or very similar values, you can't read the menu options. The Digi-View software has the same problem. Deluxe Paint changes the palette when menus are selected and restores it afterwards, so you can read the menu options.

Aegis Images and Animator files must have the proper file extension to be recognized by the file requesters in those programs. For example, all picture files must have a '.pic' extension. Only filenames with '.pic' on the end are displayed

continued...

in the file requester. Brushes (a.k.a. windows) must have '.win' extensions, color palettes must have '.col' extensions. Many users of Aegis products do not like this "feature."

Many programs do not have an "abort" button that stops printing.

According to an informal survey on People Link, the most inflexible programs when it comes to IFF are the print programs, such as Deluxe Print and Printmaster. They are not flexible about importing or exporting graphics or fonts.

Aegis Animator users complain that the program somehow disturbs the graphics chip on their machine, preventing warm-boots of the Workbench after the program has run. Users of the freely distributable Aegis Animator player program report the same problem.

Recently, a long-awaited program had a major IFF bug. Prism from Impulse was the first HAM editor for the Amiga. Unfortunately, the HAM files saved by Prism would not load into the Digi-View program. Artists had no chance of converting their HAM drawings to other resolutions using the features of Digi-View. The Liquid Light Polaroid printer would not load Prism files either. The Butcher program would load them, however. The bug was due to a malformed hunk in the Prism program. A bug fix is planned.

Prism users used a pop-up, screen-grabbing program, such as Grabbit to save screens for use in Digi-View. Other users report that programs such as Grabbit are useful for getting usable pictures from demo programs that do not allow printing or saving of your work.

IFF troubles

Can we point a finger at inflexible programs? Yes, programmers can argue about the details of implementation of an IFF program. The IFF spec is clear; it asks programs to be flexible whenever possible. In this respect, it is proper to chastise inflexible programs.

There is some evidence that original Deluxe Paint was looking at offsets and not reading IFF files according to specifications. Deluxe Paint II has replaced Deluxe Paint, and it is much more flexible about loading images of differing formats. Neither Aegis Images nor Deluxe Paint II has descriptive error messages. Deluxe Paint II responds with a box that says "I/O error" when an improper IFF file is found. Images has no error message at all.

Deluxe Paint should have a separate method of loading color palettes. Aegis Images method of color palettes makes more sense in this regard, but Images should be more flexible about loading color palettes from Deluxe Paint brushes.

Porting pictures

It is possible to transport graphic images to the Amiga from other computers. Early demos, such as RoboCity and Mandri, I were originally created on other computers. Many early pictures on IFF picture collection disks were ported from the IBM and Atari ST worlds. Electronic Arts uses the Digi-View 2.0 program to convert Amiga pictures to a form that can be displayed on the Apple II GS.

There is an Atari ST program that reads Amiga files and converts to Atari formats. There are Amiga programs that read ST formats (such as Neo, Degas and Degas Elite) and convert them to IFF. You can read Atari ST disks with the Amiga using the Dos-2-Dos program from Central Coast Software and from Commodore 64 disks with their Disk-2-Disk program. It is interesting that there is no native Amiga program to convert Amiga pictures to Atari ST format. The IFF standard has moved to the Atari ST world in the popular Degas paint program.

There are programs to convert several formats of Commodore 64 pictures to IFF format. Scott Everndon's MacView program will convert Macintosh pictures to IFF. Impulse, the makers of Prism, have in-house tools to convert MS-DOS formats from Dr. Halo, GEM and Ventura Publisher to IFF format.

One user reports there is a path to convert a picture from the Amiga to the Commodore 64, but it is very circuitous, and the colors are lost in the process. The data must travel from an Amiga to a Mac to an IBM and finally, into the Commodore 64.

IFF development

If you would like more detail about the IFF standard, look for Fred Fish disk 64. This is a copy of the official Commodore IFF disk, updated as of March 25, 1987. Carolyn Scheppner of Commodore-Amiga Technical Support (CATS) supervises the IFF standard.

A printed version of the IFF spec is available, for a small fee, from CATS as well. This manual includes the pictures that aren't available in the electronic form of the manual found on the disk. The 1985 version of the IFF spec is contained as an addendum in Volume Two of the Amiga ROM Kernal manual. The IFF spec disk includes a program called IFFCheck that verifies a file is IFF. Fish disk 38 has a similar program called IFFDump, by Matt Dillon, which disassembles IFF files for study in hexadecimal. These programs can be used to determine the resolution and size of a picture, if you are somewhat familiar with IFF terminology.

The latest IFF spec includes new hunks registered by New Horizons Software for Flow, their outline processor, and ProWrite, their word processor. The RGB4 and COMP hunks are also used in Prism files. Other new hunks are in the works for future animation and three-dimensional modeling systems. One new IFF type will store animations in a compressed form. The new IFF standards are debated among Amiga developers in the "amiga.dev/iff" conference on BIX. When the standards are approved, the additions to the manual are posted to several networks, including Usenet.

Scheppner recommends Amiga developers base their IFF routines on the code found on the IFF disk and resist the temptation to make "quick and dirty" IFF readers. "There are some subtle 'gotchas' that can cripple home-brew readers and writers," she said, such as odd-length hunks and implementing the byte-run compression scheme used to make pictures as small as possible on disk. Developers should be sure to check their programs against all other IFF programs that an Amiga owner might use in tandem with their own programs, to load and save data of all types and insure that there are no compatibility problems.

Wishes

While composing this article, I had a number of wishes and daydreams about IFF and graphics programs. Wouldn't it be great to have a color requester that used English descriptions of colors, instead of numbers? For example, you could click on a series of gadgets labeled "grey," "blue," "red," etc., as well as modifiers "dark," "light" and "ish." By clicking on a series of these boxes, you would build up a color description, such as "reddish grey" or "dark yellowish brown" and the program would give that color. This system seems much more intuitive than numbered sliders.

It would be nice if someone ported more vector-generated fonts to the Amiga. There is a large collection of public domain fonts called the Hershey fonts. A vector-generated font is described as lines, lines drawn from here to there, instead of groups of dots like the fonts on the Amiga. It should be possible to convert the Hershey fonts to raster-based dot fonts. Also, programs like Animator should accept vector fonts in script files, so you could animate letters very easily.

I know a lot of Amiga writers, both journalists and documentation writers, who would love a program that really captures a complete screen, even if the right mouse button is down-- so menus are captured as well. This program might even add the mouse pointer to the saved IFF picture for realism's sake. Normally, the pointer is not included in an IFF screen capture because it is a sprite. Sprites are a hardware trick; they are not a direct manipulation of screen memory. This program would make it easier to document a program or describe it in a review.

Many video people are begging for programs that use overscan, but there are few examples of using overscan available to developers. It is possible to use Intuition menus on overscan screens. This feature is present in some new Aegis products. If they don't have anything to do someday, Commodore should supply working code examples of this. Overscan will change the appearance of programs and scripts written for Aegis Animator because it marks off the screen with the upper left corner as the origin, so entire animations would be shifted, by a small amount, if it used overscan.

continued...

IFF is meant to be a flexible and forward-looking standard. To facilitate this, any developer can register a new IFF hunk. To be cynical, this means any developer can create a new "standard" on a whim. We must hope that developers will not abuse this feature of the IFF standard and, instead, work closely with other developers to create commonly accepted file interchange standards.

Developers should check their programs against every other IFF program they can find. This would seem to be an obvious part of the beta testing of any IFF program, but apparently, a lot of big developers missed it. Also, they should use the "CAMG" hunk to store the ViewMode and ViewPort settings, so other programs know the resolution of this image. See Carolyn Scheppner's examples of "Display.c" and "ScreenSave.c" on Fish disk 64 for more information about how to handle "CAMG".

Thank you

I would like to thank the people who contributed ideas to this article, including: Carolyn Scheppner of Commodore-Amiga Technical Support, Fred Wright and Mitsu Hadeishi on the Well, Doug Smith, Jennifer Jarik for inspiring this article, Harv Laser, Darryl Huneycutt, Jack Ungerleider, John Hoffman, Ralph Bullowa, Martin Brown, Robert Woell and Lane Winter of People Link and Larry Phillips on Compuserve.

I would like to thank the following developers who took time from their busy schedules to talk to me, as well: Stan Kalisher of Impulse, Inc., Jim Kent of Dancing Flame, Tim Jenison and Paul Montgomery of New Tek, Michelle Mehterian and William Volk of Aegis Development, Jerrel Nicholson of Eagle Tree Software and Rick Ross of Discovery Software.

Programs mentioned in this article

There are several public domain programs useful in converting IFF images. MacView by Scott Everndon works with Macintosh pictures. It can load MacPaint pictures and convert them to IFF and vice versa. It can be found on Fish disk 32 and 35. A newer version is in the public domain. The programs to convert Atari files are available on several bulleitn boards and networks.

Commodore 64 pictures can be converted to IFF with the programs on AMICUS 11. Pictures from Doodle, Koala Pad, NewsRoom and PrintShop can be converted. Apple II pictures can be converted to IFF with the programs on AMICUS 12.

•AC•

Commercial Products

Grabbit *screen capture and print program*
Discovery Software
262 South 15th Street Suite 300
Philadelphia, PA 19102
(215) 242-4666

Digi-View 2.0 *video digitizer, image processing*
Digi-Paint *HAM paint program*
New Tek
701 Jackson Suite B3
Topeka, KS 66603
(913) 354-9332

Dos-2-Dos *Reads and writes MS-DOS disks*
Disk-2-Disk *Reads and writes Atari ST disks*
Central Coast Software
268 Bowie Drive
Los Osos, CA 93402
(805) 528-4906

Aegis Animator &
Aegis Images *animation and paint programs*
Aegis Development
2115 Pico Boulevard
Santa Monica, California 90405
(213) 392-9972

Deluxe Paint II
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171

Graphicraft 1.1
Commodore Business Machines
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

Butcher *Image processing software*
Eagle Tree Software
PO Box 164
Hopewell, VA 23860
(804) 452-0623

Prism *HAM paint program*
Impulse, Inc.
6870 Shingle Creek Parkway
Minneapolis, MN 55430
(800) 328-0184

Graphicraft 1.1

Screen formats
- Low res only

Brushes
- The current set of custom brushes could be saved.
- No provision for DPaint-style brushes.

Loads
- IFF ILBM in current format
- Pictures are converted to present resolution, if possible.
- High res pictures are loaded showing only the top corner.
- Interlace loaded as top half only.
- Medium res loaded as long, wide picture.

Saves
- IFF ILBM in low res

Comments
- Loading any picture takes a long time.
- It requires an icon for any picture file to be loaded.
- Has a nice method of color cycling, using a hunk called "CCRT," for "color cycling range and timing." This gives bidirectional color cycling, as well as independent durations in seconds and microseconds.
- This program demonstrates multiple selects on menus. While the right button is held down over a menu, the left button can select items from the menu without releasing the right button.
- From the Workbench, using extended select to choose several picture icons with Graphicraft gives a slideshow of the selected pictures.

Original Deluxe Paint

Screen formats
- Low, medium and high res

Brushes
- Brushes have CMAP and BODY hunks.

Loads
- IFF ILBM in current format only
- If you try to load a picture as a brush, it gets trashed.
- It cannot load a Graphicraft brush set.

Saves
- IFF ILBM in current format only.

Comments
- Deluxe Paint needs to be converted with the FixHunk 2.0 program, if you have expansion memory.
- If you try to load a picture with a different resolution than the current resolution, it says "wrong format."
- Deluxe Paint can only be started from the CLI. Command line options select the resolution.

Deluxe Paint II

Screen formats
- Low res, interlace, medium, high res, overscan

Brushes
- Brushes have CMAP, BODY and CRNG hunks.
- If you try to load a picture as a brush, it is OK.

Loads
- IFF ILBM in any format.
- The loaded picture is not adjusted to the current format.

Saves
- IFF ILBM in current format

Comments
- Can switch resolutions without leaving the program.
- Deluxe Paint II added interlace mode, overscan mode and options for creating pictures larger than the standard resolutions and pictures larger than the screen.
- Reducing number of colors in a picture by changing resolution will recompute a new palette. This is not quite the same as what Digi-View 2.0 does.
- Brushes have color cycling info in them, using the "CRNG" hunk. Deluxe Paint brushes do not.
- Objects have a unique "DPPV" hunk which stores the current perspective plotting settings.
- It can load a Graphicraft custom brush set as a long, tall brush.

Aegis Images 1.2G

Screen formats
- Low res and medium as separate programs

Brushes
- Brushes have CMAP, BODY hunks.

Loads
- IFF ILBM in current format

Saves
- IFF ILBM in current format
- Color palettes
- Brushes

Comments
- They depend on filename extensions, '.pic' and '.hpic' for pictures, '.win' and '.hwin' for brushes, '.col' for color palettes. If any files of this type aren't present, then the file requesters won't even pop up. You have to use Aegis-conforming names, if moving files between Deluxe Paint and Images.
- Color palettes for Aegis products are just CMAP hunks, no BODY hunk. These do not load as brushes into Deluxe Paint. It says "Bad IFF form," so brushes must be used to move palettes between programs.
- Images loads Deluxe Paint brushes from any resolution and adjusts them to the current resolution.

Digi-View 2.0

Screen formats
- Low res, interlace, medium, high
- Low res HAM, interlace HAM

Loads
- Any IFF picture, including HAM, into current selected format.

Saves
- IFF HAM ILBM "4096 plus" mode
- IFF HAM ILBM, with full color palette
- IFF in any selected number of colors.
- IFF "DGVW", a unique IFF hunk. This is a format for 21 bits of color per pixel, and includes extra in formation about the color sliders in the Digi-View program.

Comments
- Can load larger than screen images into selected resolution.
- Very old Digi-View pictures weren't IFF.
- Digi-View 2.0 introduced "4096+" mode that cleans up most HAM streaks.
- This adds a color palette to a HAM picture.
- If try to load bad IFF, does nothing, no warning.
- Needs extra memory to do interlace.

Aegis Animator

Screen formats
- Low res only

Loads
- Low res IFF pictures as backdrops, brushes, masks

Saves
- Current screen as IFF
  Brushes and single-bit-plane masks

Comments
- Not really a paint program.

Butcher

Screen formats
- Displays all formats.

Loads
- Low, medium, interlace, high res and HAM

Saves
- Low, medium, interlace, high res and HAM

Comments
- Converts between any resolutions.
- Does a simple HAM-to-low res conversion.
- Doesn't load pictures larger than the screen.
- Does load buggy Prism HAM ILBM.

Prism

Screen formats
- Low res HAM

Loads
- Loads any resolution IFF, adjusts to low res HAM.

Saves
- IFF HAM, with no color palette.
- IFF HAM "RGB4", a unique IFF hunk. This is a format for 12 bits of color per pixel, compressed with Huffman encoding to save space.

Comments
- Known bug in saved IFF format.
  If picture loaded is larger than screen, picture is reduced to quarter-screen image.

Digi-Paint

Screen formats
- Low res HAM and low res interlace HAM

Loads
- Loads any resolution, but unless interlace is se lected, pictures of opposite height are squished or truncated. Interestingly, it converts medium res to low res, so that fonts are dithered and still somewhat readable in low res.

Saves
- IFF ILBM HAM with color palettes

Comments
- Pre-release beta version was viewed.

Amazing Stories
and the Amiga™

An Interview with Amazing Video Wizard, Jeff Bruette

by John Foust

In late December, an episode of television's Amazing Stories told the tale of a man who transferred his consciousness to a computer. In "The Eternal Mind," the actor appeared on a video screen in the laboratory, after the transition. The video image was distorted, as if the man had been digitized. As the man's memories faded, his image changed colors and became composed of larger and larger pixels.

Amazingly enough, the Amiga was at the heart of much of the video special effects in the episode. The technical director of the episode was Jeff Bruette, a former Commodore employee.

When Amazing Stories art director Richard Lewis wanted to use an Amiga in an episode, he was referred to Aegis Development. Lewis had been using the Amiga and Aegis Draw for set design. Bruette was working for Aegis at that time, so he was assigned to the project.

With two Amigas, a prototype Amiga Live! digitizer, a Genlock and his own Fairlight CVI video synthesizer, Bruette transformed the image of a live actor to the computerized image on the wall-sized projection screen.

The first Amiga was equipped with the Amiga Live! digitizer. A video camera's image was sent to the Live! digitizer. The resulting image was sent to the Fairlight video processor. The Fairlight is a fairly expensive video special effects computer. In this case, it produced the large-tiled pixel effect when the man's memories faded.

The Fairlight output was sent to the second Amiga through the Genlock video synchronizer. The second Amiga superimposed text over the image of the actor. A trackball was also used to control the cycling of colors in the graphics.

The resulting video image was sent to a projection system on stage - a General Electric Talaria light valve system. This system provided the exceptional brightness needed for studio work. While the Amigas and Fairlight were creating the image of an off-stage actor, actors on the main stage were interacting with the image on the projection screen.

The GE system also has a high level of persistence, so the video image stayed on the screen long enough to be filmed, without the usual distortion of roll bars when a video screen is filmed.

As a graphics-oriented company interested in the Amiga, Aegis Development coordinated the Amazing Stories project. According to Bruette, it was a good public relations opportunity for them. Bruette left Aegis at the end of October, a few weeks after completion of the Amazing Stories production.

Bruette was capable of doing the Amazing Stories episode himself, because he owns his own video equipment. In fact, after he left Aegis, Bruette formed his own production house, Prism Computer Graphics. At this point, Prism is a one-man company, but may expand in the future. "At this point, it is just me, but at the rate that it's going, in the next three or four months, I'll have to hire somebody," Bruette said.

The story of Bruette and the Amazing Stories episode was also featured in the February 1987 Computer Graphics World magazine, a well-respected computer graphics industry publication. The article, titled "Amazing Amigas," was written by Jennifer Dritt Wright.

Bruette and Commodore

Bruette left Commodore Amiga in late May 1985, just prior to the layoffs. He had no prior knowledge of the layoffs. By staying an extra week, Bruette saved someone else from being fired, he said.

Bruette worked for Commodore when the VIC 20 was popular. He was hired to do graphics for video games. In the Commodore 64 world, he did "Wizard of Wor." Bruette's official title was "Special Applications," which meant he moved between projects involving computers and graphics. He did custom graphics for Commodore commercials and computer packaging. When the Amiga came along, he was in sales under Frank Leonardi.

Bruette also guided development of several products at Island Graphics. At that time, Island was developing Graphicraft, which later became Aegis Images. Island Graphics has coordinated custom graphics software for many companies, including Aegis Development. Because of this connection, Bruette had a path to move to Aegis, after leaving Commodore.

"What was going to be Graphicraft is now known as Aegis Images. The Aegis products were originally going to be Commodore products. During the development of those

continued...

products, I was at Island Graphics, overseeing their creation of them. That is why I looked at Aegis for my first job, after I left Commodore. It seemed an obvious route to take."

I spoke with Bruette at the Consumer Electronics Show in Las Vegas in early January 1987.

AC: What was it like to interact with the production staff of Amazing Stories?

JB: "It was a little difficult because the art director knew the capabilities of the machine. Having used the machine for a while, he knew that things didn't just happen overnight. But, in a production like that, that runs $55,000 a day, sometimes overnight isn't even soon enough.

"He would design a look that he wanted digitizing at this point, with lots of technobabble superimposed. We would get on the set and go through rehearsal, and the director would say, "There is too much going on on the screen. Change this, move that." We would have three or four minutes to rush and change. That was the most difficult part of it."

AC: What sort of tools did you use?

JB: "All the technical graphics were done in BASIC. It was more operation of the equipment than creation of special software. The only special software were BASIC programs to display countdowns, things like that." For the live actor sequences, Bruette used the standard Amiga Live! software.

AC: Did anyone on set become interested in the Amiga?

JB: "There were three people. The video coordinator made sure the video feeds were going where they should and he would adjust the video levels. The machines that were used as props on the shows were supplied by the [Amiga] representative in that area, Marketshare. There were a total of three machines used as props.

Of those machines, all three were sold by the end of the production. The people involved in the show were so impressed that they bought the machines. These three people included one of the production coordinators. It had nothing to do with Amazing Stories props; they wanted them for themselves. One of the directors might have bought one."

AC: Why were the Amiga logos on the computers masked out?

JB: "On a TV show, it is very rare that you see a name brand on a product. Franklin or IBM doesn't want to be the middle of an episode of Amazing Stories where they've been showing Amigas, and panning across the Amiga logo, and then breaking for one of their [Franklin or IBM's] commercials. But when it comes to movies, they will almost always leave the name visible."

Bruette explained that, in movies, companies often pay for the privilege of displaying their names. On television, producers are concerned with offending advertisers, so names are masked out with tape (as they were in the Amazing Stories episode).

AC: What are the limits of the Amiga, as a real video tool?

JB: "The non-standard video quality. They used very poor RGB to composite conversion hardware. It is borderline at best, for professional uses. Usually, you have to take RGB output and convert it to composite externally. Even then, it is not the best it could be.

Bruette also discussed his preference of drawing programs and the features that he likes most.

"I would sincerely hope that future Amigas would have better composite video output. Besides that, I have found very little that it is not capable of doing. Usually, if you can't accomplish something with it one way, you can do it another. If [Aegis] Images can't to it, then Deluxe Paint will. Between Deluxe Paint and Images, I prefer Images, between Images and Paint II, I prefer Deluxe Paint II.

"In the animation side of things, I definitely prefer [Aegis] Animator over Deluxe Video because, in my applications, I don't have to be involved in synchronizing to sound or anything like that.

There is nothing I can see that needs to be added to Deluxe Paint II, unless there was some way of creating cell animation, so that you could create a man running, so that you could create five or six running men and see them running without leaving the program. It is quite possible that Aegis will do it in a future paint program, since they are doing more work on animation."

AC: How useful are the video overscan abilities of Deluxe Paint II?

JB: "Overscan was one of those things that everyone complained about when it wasn't there, and now that its available, nobody cares. Let's find something else to complain about now. I felt it was a limitation. But now that Deluxe Paint supports it, I haven't used it. I take more advantage of the full-page mode. I don't see anything that the Amiga really needs, besides a little cleanup on the software."

Star Trek IV

When the latest Star Trek movie came to the theaters, it was accompanied by a rumor on the Amiga community grapevine that the Amiga was to have been the featured computer in the movie. Apparently, a Commodore screw-up prevented the Amiga's big screen debut. Bruette explained the story behind the rumor.

At the time Star Trek IV was being produced, Bruette was still with Commodore. Commodore applied to a product placement agency, in order to have the Amiga appear in a motion

picture. Similar to an advertisement, the placement agency would coordinate the appearance of Commodore computers in a movie. Bruette was told to keep his schedule open for three weeks, in case a movie deal came up for the Amiga.

When the three weeks passed, Bruette was told that the project was shelved because another computer company paid more money than Commodore for the Star Trek appearance. The placement agency would not say which company outbid Commodore.

At the same time, a Star Trek video engineer sent a letter to Commodore. In the final production, the engineers can decide which tools are to be used, and someone wanted an Amiga. Bruette thinks the letter was routed to Commodore technical support... and a reply soon returned.

According to Bruette, the reply stated, "We don't give out computers, but we'll get you one at developer price." But the engineeer persisted. "He wanted one bad enough that he sent them a check, with a letter saying 'I want one, but I need it right away. We are going into production.'" His check was returned to him, with a letter saying "I'm sorry, we have to take orders as they are received. We cannot bump your order ahead for any reason at all. So, send your check back if you want to wait."

In the end, the Apple Macintosh was used in one highly-visible scene in the movie; but Bruette clarifies where the Amiga was supposed to appear: "But that wasn't where the Amiga was going to be used. All the technical-type scenes, not as a prop. The video feeds were going to be Amiga stuff. They ended up using high-end Calcomp systems. A lot of it was actually animation, hand-drawn on paper, that was then videotaped."

Actually, another Commodore computer long preceeded the Amiga in the Star Trek movie series. In a scene in Kirk's home in Star Trek II, as the camera panned over his antique collection, a Commodore PET computer sat happily blinking its cursor.

Bulletproof

Bruette is now up against similar problems with Commodore, with a movie called 'Bulletproof.' The story involves a high-tech tank. The tank has computer displays that would be created with the Amiga. However, in order to film this, all the displays need to be synchronized. Only Genlock can do this. Bruette has had trouble getting the necessary Genlock devices from Commodore:

"The Genlocks do exist. But Commodore told me they can't supply them, until they fill all orders and all back-orders. I understand that the customers need them, but in the case of using two units being used like this, would surely outweigh two customers waiting an extra couple weeks for Genlock, now that they've waited well over a year for it. Some of these dealers don't even know what they are ordering. They will get them and they will sit on the shelves for a couple weeks anyway.

"I don't think that, in most cases, I shouldn't be able to call Commodore and say 'Send me this' or 'Send me that,' because they've done too much of that in the past and they've gotten abused in that situation. When there is an honest need, there should somebody that looks at that, and makes these sorts of decisions. To have a computer used on a movie or a television show is worth thousands of advertising dollars."

Oingo Boingo

Bruette also works with rock group Oingo Boingo and their lighting director Charlie Unkeless. The band uses the Amiga to create animations and graphics to project on large screens along with other video images. Boingo is now considering using Amiga to do music with Soundscape.

The band has two Amigas, one for the video production on the road and one for the office. Unkeless uses the Amiga to organize the Oingo Boingo fan club and order the band's accounting.

•AC•

CSA's Turbo Amiga

Amazing Reviews...

CSA's Turbo-Amiga Tower

Reviewed by Alfred A. Aburto

The ***Turbo-Amiga Tower*** expansion chassis brings Amiga 2000 capability to the Amiga 1000. The Tower, a product of San Diego's ***Computer System Associates***, runs a Motorola 68020 32-bit CPU, a 68881 floating-point math processor and 32-bit memory at a clock speed of 14.32 MHz.

The Turbo-Amiga Tower is nearly identical to CSA's earlier Amiga 1000 Turbo-Amiga. All the features of the original Turbo-Amiga are included in the new and improved Turbo-Amiga Tower. The only difference lies in the redesign of the PC boards and chassis to fit the Amiga 2000 PC board specifications. Additional PC board slots have been added and the maximum 32-bit memory capacity is increased to 12 megabytes.

The Tower contains its own power supply, has room for seven Amiga 2000 Zorro bus compatible PC boards and can accomodate two 20 or 40 Megabyte fixed hard disk units (one of which can be configured to be removable). The Turbo-Amiga Tower plugs into the 86-pin expansion port on the side of the Amiga 1000.

Such enhancements are simple on the Amiga 1000 because of Commodore-Amiga's open architecture policy. More important, the Amiga's multitasking operating system is designed to operate with both the 68020 and 68881.

Operating at 14.32 MHz, with 32-bit memory, helps the 68020 processor provide up to a 5.5 improvement in processing performance over the Amiga 1000's Mhz 68000. In mathematics intensive operations, the Turbo-Amiga flaunts processing speed performance improvements of a factor of 500 over the Amiga 1000.

The Turbo-Amiga lifts numeric processing capabilities close to the level of the best VAX-8600 systems (comparable to an HP-9000/320 or Sun-3/160 system). The system is by no means equivalent to a VAX-8600 system, but can provide numeric processing speed and accuracy nearly equal to a VAX-8600.

The future shines bright for the Turbo-Amiga Tower. Upgrades to even higher performance Motorola processors (such as the 68030 and 68882) will be relatively easy, at clock speeds up to 25 MHz. Even now, select MC68881 coprocessors can operate at a clock speed of 28.64 MHz with the Turbo-Amiga. The result—a system with expectional numeric processing capability.

Chassis design

The Turbo-Amiga chassis mates with the Amiga 1000 by press-fitting its 86-pin female socket into the 86-pin expansion bus connector on the side of the Amiga 1000. Vertical alignment is steadied by four rubber foot pads. The two units can be firmly locked together to guard against accidental separation.

The Turbo-Amiga draws absolutely no power from the Amiga 1000. The Tower enclosure comes with a 200- watt power supply. Factory-wired options allow the supply to operate from line voltage of 110 or 220 VAC and 50 or 60 Hz. The enclosure also contains a whisper fan.

A back-plane, containing electronic circuitry, converts the 86-pin Amiga 1000 expansion bus signals to the 100-pin Commodore-Amiga specified "Zorro" bus standard. Seven 100-pin Zorro bus connector sockets accomodate any Zorro bus standard (Amiga 2000) PC board. The Zorro bus standard is designed for 16-bit data and 24-bit addresses.

The 68020's 32-bit data and 32-bit address signals are generated by CSA's CPU board. The signals are routed to other CSA 32-bit PC boards, via four 40-pin ribbon edge card connector sockets and cables. To minimize noise and cross-talk, all other connector pins are grounded .

CPU board

The CSA Turbo-Amiga Tower CPU board contains the Motorola 68020 32-bit microprocessor, the 68881 hardware mathematics processor and all electronic circuitry. On power-up, the board generates a signal to put the Amiga 1000's

continued...

68000 CPU in the electronic "tri-state" condition ——in effect, removing the 68000 from the system. The Turbo-Amiga's 68020 then takes control of the system.

The CPU board runs the 68020 at a clock speed of 14.32 MHz and assumes 32-bit memory is present for bus accesses in the 32-bit 68020 address range ($01000000 through $FFFFFFFF). The board switches the 68020 to a clock speed of 7.16 MHz, uses electronic circuitry to imitate the 68000 signals and assumes 16-bit memory is present for bus accesses in the normal 68000 address range ($00000000 - $00FFFFFF). This process is necessary for operation of the Amiga 1000's normal hardware (such as the graphics co-processor), designed to work specifically at 7.16 MHz.

The 68881 floating-point processor is coupled with the 68020 and the 14.32 MHz clock as a true 32-bit co-processor. The 68881 always operates at a clock speed of 14.32 MHz, even with the 68020 operating at 7.16 MHz. Code loaded into the 68020's 256-byte internal instruction cache memory will also always execute at the 14.32 Mhz clock rate. The 68020 is switched to a clock rate of 7.16 Mhz only when external 16-bit memory bus accesses are made in the Amiga 1000's normal 16-Megabyte address range ($00000000 - $00FFFFFF).

For this reason, only essential Turbo-Amiga code is loaded into the Amiga 1000's normal 16-Megabyte memory space.. For best performance, most Turbo-Amiga user programs must be in the larger 4 Gigabyte MC68020 memory space defined by CSA ($01000000 through $FFFFFFFF).

With 16-bit memory, the 68020 operates at a distinct disadvantage. Because the 68020 always does a 32-bit external memory prefetch, an extra read cycle is necessary when accessing 16-bit external memory. The 68020's 16-bit memory requires 5 clock cycles to fetch a memory instruction. The 68000, using its additional hardware to emulate the 68000 in the address range of $00000000 -$00FFFFFF, needs only 4 clock cycles. For this additional reason, Turbo-Amiga users prefer (or demand) to run their programs in 32-bit memory ($01000000 - $FFFFFFFF), in lieu of the Amiga's 16-bit memory ($00000000 - $00FFFFFF).

In my opinion, the 68881 is the finest general purpose hardware floating-point mathematics processor now available. The IEEE Standard for Binary Floating Point Arithmetic is fully implemented and functions not defined by the IEEE Standard (including a full range of trigonometric and transcendental functions) are also supported. Seven data types are supported: byte, word and long word integers, single (32-bit), double (64-bit) and extended precision (80-bit) floating-point real numbers and packed BCD string floating point numbers (Fortran "E" and "F" formats) which easily convert to ASCII for printing.

Oft used floating-point constants are available from an on-chip ROM look-up table. Table 2 shows the operations and functions available from the 68881. Some of these functions, like the 'SinCos(x)' function (which computes the Sin(x) and Cos(x) simultaneously) would make very useful additions to the Standard Mathematics Libraries. SinCos(x) would be ideal for graphics applications, such as vector and rotation operations, as it is nearly twice as fast as computing sin() and cos() separately.

Memory boards

As many as six 512K/2048K byte memory boards can be configured into the Tower enclosure for a total of up to 12 Megabytes of RAM. The CSA memory boards consist of relatively expensive 32K x 8 bit or 128K x 8 bit static RAM devices. The boards have a data access time of 100ns or faster to ensure operation at 14.32 MHz, with no wait states. The CSA memory boards accept either the 32K x 8 or 128K x 8 type static RAMs and can be arranged to occupy 32-bit addresses above $01000000.

The memory is handed over to the operating system via an assembly routine in the 'startup-sequence' file. This routine gets the 32-bit memory ready for use at system startup or reboot. High cost as compared to other 512K/2048K byte memory boards available for the Amiga is the only negative feature of the CSA 32-bit memory boards. 32-bit memory boards designed with higher density dynamic RAM devices would probably help reduce system price.

Disk Interface

The SCSI Disk Interface board provides the logic to convert the 100-pin Zorro bus to the industry standard Small Computer System Interface (SCSI) bus. The SCSI bus is a widely used multiple master/slave bus which interfaces with many industry standard devices equipped with SCSI controllers. The board's line drivers can drive as many as 8 SCSI devices. The NEC 5385E SCSI controller, along with the MC68450 DMA controller, provides full DMA access to the Amiga Zorro bus. This board's electronic circuitry tells the Amiga's operating system of the presence and operational status of devices connected to the SCSI bus.

Two 20 or 40 Megabyte fixed hard disk units can be fitted into the Turbo-Amiga Tower enclosure. The CSA fixed hard disk drive contains an SCSI controller to communicate with the Amiga's disk operating system via the SCSI Disk Interface board.

A removable 20-Megabyte hard disk, which also contains a SCSI controller, can also be fitted into the Turbo-Amiga Tower enclosure. This 20-Megabyte hard disk comes in a rugged slide-in cartridge, for quick removal from the Turbo-Amiga Tower enclosure. The unit contains integral power, a data connector socket and sturdy guide pins to mate with the Tower's bus. This removable hard disk is intended for storage of classified information which cannot be left unsecured in the computer system.

Performance

Table 3 should give you a look into the performance of the Turbo-Amiga, in relation to the Amiga 1000. I used Absoft's FORTRAN 77 V2.2B and the Manx Aztec C V3.30e compilers

for this comparison because both have the capability to generate 68000, 68020 and 68020/68881 code. The Quelo assembler also has such capability. I used four different programs for the comparision:

- The Byte Calculations program (see "Inside the IBM PCs," *Byte*, Fall 1985, page 195).
- The Savage program (see *Dr. Dobb's Journal*, 1983, page 120).
- The Sieve of Eratosthenes program (see 'Eratosthenes Revised," *Byte*, by Jim and Gary Gilbreath, January 1983, page 283).
- The Dhrystone Benchmark V1.1 from USENET.

The results in Table 3 are only an example of relative performance. Other programs, running with the same or different computer languages, will certainly produce different results. For example, TDI's Modula-2 compiler for the Amiga executes the *Byte* Calculations program in 3.6 seconds and Lattice C V3.03 will run the Sieve program in 0.8 seconds with the Turbo-Amiga.

Simply stated, Table 3 shows that the Turbo-Amiga provides a significant performance upgrade of the Amiga 1000. The Turbo-Amiga clocks in at 3008 Dhrystones/second, leading the way towards low-cost, high-performance computer systems.

Table 4 indicates how the Turbo-Amiga rates versus other computer systems. I used the 'Savage' program for this comparison. The Savage computes the statement

```
a = tan(atn(exp(log(sqr(a*a))))) + 1.0
```

in a loop from i=1 to i=2499 with 'a' initialized to 1.0. The final value of 'a' should be 2500. To obtain a meaningful final result, IEEE double precision floating-point variables must be used. IEEE single precision floating-point variables do not contain enough bits of precision to handle the atn(a) function when 'a' is 'large' (near 2499).

I used the Savage because the tan(a), atn(a), exp(a) and log(a) are among the toughest and most time consuming standard functions. These functions are approximated by polynomials and computed internally, using floating-point adds, subtracts, multiplies and divides. Because the exact result is known, I could obtain a computational error check. Table 4 shows that the Turbo-Amiga (really the 68020/68881) is an outstanding performer, (relative to the Savage program) as compared to other well-known computer systems, including the Sun-3/160, HP-9000/320 and the VAX-8600.

Table 4 also shows that C compilers are among the poorest performers when it comes to software double-precision floating-point operations. BASIC interpreters (particularly HP Basic V2.0) are up to 5 times faster than the C compilers with Savage's floating-point operations.

Even Absoft's FORTRAN 77 compiler is not as fast as AmigaBASIC, in regard to software floating point. You might expect compilers and interpreters to take nearly the same amount of time to run the Savage benchmark. After all, the compiler and interpreter use a similar machine language subroutine to compute math functions. They spend much time doing floating-point adds, subtracts and multiplies, until an approximate result for the math function is ground out.

The compiler and interpreter should take approximately the same time to grind out the result because the floating-point algorithms to do adds, multiplies or compute the math functions are pretty much standardized. In general, C compilers could be much improved in the area of double precision floating-point operations.

Software

I have been using a 68020/68881 in my Amiga for more than a year. Initially, I used a CSA piggy-back board 68020/68881 replacement for the 68000. It worked only at 7.16 MHz, but I

continued...

felt it was worthwhile because of the performance improvements provided by the 68881. The big problem was that the hardware was useless without software to support it.

The AmigaDOS 1.0 and 1.1 did not support the 68881 for multitasking operations. With these operating systems, the state of the 68881 was not saved or restored when the OS switched to another task. As a result, the system would invariably crash.

You could avoid this uncomfortable situation by using the Executive Forbid() and Permit() routines, but the real solution begged for a change in the operating system. Enter the wizards of the operating system at Los Gatos (Neil Katin, Dale Luck et al.) who solved the problem for AmiagDOS version 1.2. This improvement was an important step in the Amiga's development—— the Amiga could now perform at a level rivaling much larger and more expensive machines.

The first compiler to support the Amiga with the CSA 68020/68881 hardware was Absoft's FORTRAN 77. I have run many Fortran programs and benchmarks to compare the performance of Turbo-Amiga and Absoft's FORTRAN compiler performance with other systems. I am always amazed - even "tickled pink" - when the Turbo-Amiga/Absoft system performs faster or nearly as fast as much larger systems.

Two other products have been designed with the Turbo-Amiga 68020/68881 specifically in mind: the Quelo 68020/68881 Macro Assembler and the 68020/68881 Manx Aztec C V3.4. The Quelo Assembler also supports the Motorola MC68851 paged memory management coprocessor. The TDI Modula-2 compiler and Lattice C also work correctly with the 68020, but do not support the 68881.

Almost all my Amiga software works fine with the Turbo-Amiga. A few problems do pop up, due mostly to software developers who have not followed the Amiga programming conventions in the Amiga ROM Kernel Manual. One convention asks that all address should be 32, rather than 24-bits. . . yet Microsoft's AmigaBASIC and other products, including the Metascope debugger, restrict operation to a 24-bit address range. AmigaBASIC simply does not work with the Turbo-Amiga's 32-bit memory. Metascope prints an 'illegal address' when attempting to access 32-bit memory. I was particularly disappointed with AmigaBASIC because it is such a useful product.

AmigaBASIC™

In an attempt to discover the problem, I investiged AmigaBASIC a little deeper. I found several routines throughout AmigaBASIC, which load a 24-bit address pointer from memory to an address register. The 24-bit address is loaded into a cleared data register using 'byte move' and '8-bit rotate left' instructions. The contents of the data register, with the upper byte cleared, are then loaded into an address register.

This procedure destroys any possibility of AmigaBASIC working with the Turbo-Amiga and 32-bit memory. Efficiency is also tossed out, since the 68000 and 68020 are capable of moving a 32-bit number from memory to an address register with only one instruction! AmigaBASIC is overdue for an overhaul. It will work with the Turbo-Amiga, but only with 16-bit memory running at 7.16 Mhz, in the normal 68000 address range. AmigaBASIC needs the ability to work with the Turbo-Amiga at 14.32 MHz with 32-bit memory.

While investigating AmigaBASIC, I uncovered all its software floating point routines. I hot-patched the program and converted these routines to 68881 instructions. Table 4 shows the results of this modification with the Savage benchmark. The 68881-enhanced AmigaBASIC ran the program in 6.8 seconds——an excellent result as compared to Microsoft Fortran on the IBM PC-AT with its floating-point coprocessor, which took 7.2 seconds.

Floating-point operations with the 68881 could make AmigaBASIC an even better product. The only reason I can find for Microsoft's outdated technique for fetching address pointers from memory is some great concern to use every available bit of memory.

AmigaBASIC's PEEK(a) instruction, where 'a' is an address pointer, presents an additional problem. The argument 'a' is treated as a signed integer. Accordingly, any attempt to PEEK data at an address above $7FFFFFFF results in an overflow error. This problem would not have occurred if Microsoft had adhered to Commodore-Amiga's software design convention that all address pointers should be 32-bit unsigned integers. We can work around this problem, but it should not have been a problem in the first place.

The True Basic compiler also fails with the Turbo-Amiga's 68020. I could not run a True Basic program without crashing the system. I think the problem here was that True Basic used the 'move status register to effective address instruction' (MOVE SR,<ea>). Such a move is OK with the 68000 in the 'user state,' but is a privileged instruction violation with the 68020. This instruction should be avoided to ensure software upward compatibility to Motorola's other processors. Rather, a 'MOVE CCR,<ea>' should be used in the user mode.

Conclusion

Most Amiga software I've examined works with the Turbo-Amiga's 68020. Only a few software products directly support the 68881, but the basic tools (several compilers and an assembler) are available.

A 68881 math library for the Amiga could drastically simplify 68881 interfacing. Such an option would allow any software product (compiler, assembler or whatever) to take advantage of the 68881. The capability to move the Amiga's Operating System from 16-bit memory to the high speed 32-bit memory of the Turbo-Amiga systems could also greatly enhance Turbo-Amiga performance.

Overall, the CSA Turbo-Amiga and new Turbo-Amiga Tower are leading the way towards the ultimate combination of high performance and relatively low cost in computer systems.

•AC•

CSA Turbo-Amiga Products as of May 87

1)	Turbo-Amiga Tower Chassis With 7 Zorro Bus Slots. 4 IBM PC-AT Slots. 1 CSA 68020/68881 CPU board Slot.	$1095
2)	68020/68881 Zorro Bus CPU board, 32-Bit Address and Data Bus, 14.32 MHz.	$1480
3)	512K 32-Bit Memory board (Static RAM)	$1295
4)	2 Meg 32-Bit Memory board (Static RAM)	$3995
5)	SCSI Controller board	$695
6)	20 Meg Fixed Hard Disk Unit	$910
7)	40 Meg Fixed Hard Disk Unit	$2200
8)	20 Meg Removable Hard Disk Unit	$2495

MC68881 Functions (Typical Run Times)

Function	Description	Clock Cycles	Time (µsecs)
1) FABS	Absolute Value	60	4.2
2) FACOS	Arc Cosine	650	45.4
3) FADD	Addition	78	5.4
4) FASIN	Arc Sine	606	42.3
5) FATAN	Arc Tangent	428	29.9
6) FATANH	Hyperbolic ArcTan	718	50.1
7) FCMP	Compare	60	4.2
8) FCOS	Cosine	416	29.1
9) FCOSH	Hyperbolic Cos	632	44.1
10) FDIV	Divide	130	9.1
11) FETOX	exp(x)	522	36.5
12) FETOXM1	exp(x) - 1.0	570	39.8
13) FGETEXP	Get Exponent	70	4.9
14) FGETMAN	Get Mantissa	56	3.9
15) FINT	Int. (Floating Point)	80	5.6
16) FINTRZ	Int. (Round To Zero)	80	5.6
17) FLOGN	log(x)	550	38.4
18) FLOGNP1	log(x + 1.0)	596	41.6
19) FLOG10	log10(x)	606	42.3
20) FLOG2	log2(X)	606	42.3
21) FMOD	Modulo Remainder	606	42.3
22) FMOVE	Move To Register	58	4.1
23) FMOVE	Move To Memory	86	6.0
24) FMOVECR	Move Const Fr ROM	29	2.0
25) FMUL	Multiply	98	6.8
26) FNEG	Negate	60	4.2
27) FREM	IEEE Remainder	92	6.4
28) FSCALE	Scale Exponent	68	4.7
29) FSGLDIV	Single Precision Div.	96	6.7
30) FSGLMUL	Single Precision Mult.	86	6.0
31) FSIN	Sine	416	29.1
32) FSINCOS	Simult. Sin & Cos	476	33.2
33) FSINH	Hyperbolic Sine	712	49.7
34) FSQRT	Square Root	132	9.2
35) FSUB	Subtract	78	5.4
36) FTAN	Tangent	498	34.8
37) FTANH	Hyperbolic Tan	686	47.9
38) FTENTOX	10 ** x	592	41.3
39) FTST	Test	58	4.1
40) FTWOTOX	2 ** x	592	41.3

Performance of Turbo-Amiga Relative to Amiga 1000

System CPU / FPP	Clock (MHz)	Cache (Bits)	Mem.	Time (Sec)	Performance (68000/68020)
Sieve of Eratosthenes (Absoft FORTRAN 77 V2.2B Compiler)					
Amiga (68000/-----)	7.16	---	16	6.6	---
Turbo-Amiga (68020/-----)	7.16	off	16	6.5	1.0
Turbo-Amiga (68020/-----)	7.16	on	16	3.1	2.1
Turbo-Amiga (68020/-----)	14.32	off	32	1.7	3.9
Turbo-Amiga (68020/-----)	14.32	on	32	1.2	5.5
Savage Benchmark (Absoft FORTRAN 77 V2.2B Compiler)					
Amiga (68000/-----)	7.16	---	16	77.2	---
Turbo-Amiga (68020/-----)	7.16	off	16	74.0	1.0
Turbo-Amiga (68020/-----)	7.16	on	16	49.0	1.6
Turbo-Amiga (68020/-----)	14.32	off	32	24.3	3.2
Turbo-Amiga (68020/-----)	14.32	on	32	21.3	3.6
Turbo-Ami (68020/68881)	7.16	off	16	0.42	184.0
Turbo-Ami (68020/68881)	7.16	on	16	0.40	193.0
Turbo-Ami (68020/68881)	14.32	off	32	0.40	193.0
Turbo-Ami (68020/68881)	14.32	on	32	0.40	193.0
BYTE Calculations (Absoft FORTRAN 77 V2.2B Compiler					
Amiga (68000/-----)	7.16	---	16	4.4	---
Turbo-Amiga (68020/-----)	7.16	off	16	4.8	0.9
Turbo-Amiga (68020/-----)	7.16	on	16	1.9	2.3
Turbo-Amiga (68020/-----)	14.32	off	32	1.4	3.1
Turbo-Amiga (68020/-----)	14.32	on	32	1.1	4.0
Turbo-Ami (68020/68881)	7.16	off	16	0.14	31.4
Turbo-Ami (68020/68881)	7.16	on	16	0.13	33.8
Turbo-Ami (68020/68881)	14.32	off	32	0.13	33.8
Turbo-Ami (68020/68881)	14.32	on	32	0.12	36.7
Dhrystone Benchmark (Manx Aztec C V3.30E)					
Amiga (68000/-----)	7.16	---	16	46.2 (1083)	---
Turbo-Amiga (68020/-----)	7.16	off	16	54.5 (917)	0.9
Turbo-Amiga (68020/-----)	7.16	on	16	43.1 (1169)	1.0
Turbo-Amiga (68020/-----)	14.32	off	32	18.3 (2729)	2.5
Turbo-Amiga (68020/-----)	14.32	on	32	16.6 (3008)	2.8

Number of Dhrystones/second shown in parentheses

The following programs were used for this comparison:

1) The Byte Calculations prog.("Inside The IBM PC's", *Byte*, Fall 85, pg.195.)
2) The Savage prog.(*Dr. Dobb's Journal*, 1983, pg.120.)
3) The Sieve of Eratosthenes prog.("Eratosthenes Revised", *Byte*, Jan. 83, pg. 283.)
4) The Dhrystone Benchmark V1.1 from USENET

The Savage Benchmark Results

System	CPU / FPP (MHz)	Clock	Language	Time (sec) (a-2500)	Error
VAX-8600	----------	-----	VAX Fortran (VMS 4.4)	0.28	6.6 E-10
Turbo-Amiga	(68020/68881)	14.32	Absoft F77 V2.2B	0.39	2.7 E-12
VAX-8650	----------	-----	VAX Fortran (VMS 4.4)	0.40	6.6 E-10
Sun-3/160	(68020/68881)	16.67	Sun 3.0 F77	0.4	2.0 E-12
Turbo-Amiga	(68020/68881)	14.32	Lattice C/68881 Assem	0.46	9.2 E-13
Turbo-Amiga	(68020/68881)	14.32	Manx Aztec C V3.30E	0.57	1.2 E-09
HP 9000/320	(68020/68881)	-----	Fortran 77	0.7	3.2 E-09
HP 9000/320	(68020/68881)	-----	Pascal	0.7	2.8 E-07
VAX-11/785	----------	-----	VAX Fortran (VMS 4.3)	0.77	6.6 E-10
Amiga	(68020/68881)	7.16	Absoft F77 V2.2B	0.78	2.0 E-12
VAX-8600	----------	-----	Fortran 77	0.9	1.8 E-08
Turbo-Amiga	(68020/68881)	7.16	Lattice C/68881 Assem	0.92	5.9 E-12
HP 9000/320	(68020/68881)	-----	C	1.0	2.5 E-08
VAX-11/780	----------	-----	VAX Fortran (VMS 4.4)	1.16	6.6 E-10
DEC 2060	----------	-----	----------	1.6	2.0 E-12
VAX-11/750	----------	-----	Fortran 77	1.9	6.6 E-10
Masscomp	(68010/ FPP)	-----	----------	2.1	3.2 E-07
VAX-11/780	----------	-----	UNIX 4.3BSD (F77-O)	2.7	1.8 E-12
Intel 80386	(80386/80287)	16.00	Phoenix BIOS	2.75	--------
DMS	(8086/ 8087)	-----	Turbo Pascal	3.8	1.1 E-09
Zenith Z-248	(80286/80287)	8.00	MS Fortran77 V3.20	4.5	1.2 E-09
IBM PC-AT	(80286/80287)	6.00	ProFor F77	4.9	8.7 E-11
Turbo-Amiga	(68020/68881)	7.16	MS AmigaBASIC V1.2	6.8	1.2 E-09
IBM PC-AT	(80286/80287)	6.00	Microsoft Fortran 77	7.2	1.2 E-09
IBM PC-AT	(80286/80287)	6.00	Turbo Pascal	7.4	1.2 E-09
IBM PC	(8088/ 8087)	4.77	Microsoft C	8.0	1.2 E-09
Turbo-Amiga	(68020/68881)	7.16	AmigaBASIC (Cache Off)	8.8	1.2 E-09
Turbo-Amiga	(68020/-----)	14.32	Absoft F77 V2.2B	21.3	1.8 E-07
Sun-3/160	(68020/-----)	16.67	Sun 3.0 F77	21.5	3.1 E-07
Turbo-Amiga	(68020/-----)	14.32	Absoft F77 (Cache Off)	24.3	1.8 E-07
Turbo-Amiga	(68020/-----)	14.32	Manx Aztec C V3.30E	34.0	3.2 E-07
HP 9826	(68000/-----)	8.00	HP Basic V2.0	44.5	3.2 E-07
Turbo-Amiga	(68020/-----)	7.16	Absoft F77 V2.2B	49.0	1.8 E-07
Turbo-Amiga	(68020/-----)	14.32	Lattice C V3.03	55.4	3.2 E-07
IBM PC-XT	(8088/ 8087)	4.77	Gauss	58.0	1.2 E-09
Turbo-Amiga	(68020/-----)	7.16	MS AmigaBASIC V1.2	59.0	3.2 E-07
HP Integral	(68000/-----)	-----	Basic Interpreter	60.9	3.2 E-07
HP Integral	(68000/-----)	-----	C	63.0	3.2 E-07
Amiga	(68000/-----)	7.16	True Basic (Compiler)	65.2	3.0 E-03
Turbo-Amiga	(68020/-----)	7.16	AmigaBASIC (Cache Off)	67.0	3.2 E-07
Atari 520ST	(68000/-----)	8.00	Absoft F77 V2.2	67.6	1.7 E-07
Turbo-Amiga	(68020/-----)	14.32	Lattice C (Cache Off)	72.4	3.2 E-07
Atari 520ST	(68000/-----)	8.00	Alycon C V4.4	72.6	1.8 E-07
Amiga	(68000/-----)	7.16	MS AmigaBASIC V1.2	73.0	3.2 E-07
Turbo-Amiga	(68020/-----)	7.16	Absoft F77 (Cache Off)	74.0	1.8 E-07
Amiga	(68000/-----)	7.16	Absoft F77 V2.2B	77.2	1.8 E-07
Atari 520ST	(68000/-----)	8.00	Mark Williams C V2.0	82.7	1.2 E-07
Atari 1040ST	(68000/-----)	8.00	Mark Williams C V1.1	83.7	1.2 E-07
Turbo-Amiga	(68020/-----)	7.16	Manx Aztec C V3.30E	91.4	3.2 E-07
HP Integral	(68000/-----)	-----	Absoft F77	100.0	1.8 E-07
Atari 520ST	(68000/32081)	8.00	Megamax C	119.0	2.2 E-08
Amiga	(68000/-----)	7.16	Manx Aztec C V3.30E	120.3	3.2 E-07
Turbo-Amiga	(68020/-----)	7.16	Lattice C V3.03	139.0	3.2 E-07
Macintosh	(68000/-----)	7.83	MAC C	221.0	-------
Amiga	(68000/-----)	7.16	Lattice C V3.03	234.0	3.2 E-07
Macintosh	(68000/-----)	7.83	DeSmet C	244.0	--------
Commodore-128	(8502/-----)	2.00	Basic Interpreter	256.0	9.0 E-04
Macintosh	(68000/-----)	7.83	Manx Aztec C	353.0	--------
Atari 520ST	(68000/-----)	8.00	Megamax C	495.0	8.5 E-07
IBM PC-XT	(8088/-----)	4.77	Basica V2.10	895.0	3.0 E-08
Tandy PC-5	----------	-----	Basic Interpreter	961.0	2.7 E-03

Notes:

(1) The Savage Benchmark, by Bill Savage, first appeared in Dr Dobb's Journal, Sept. 1983, Page 120.

(2) Many people, on various computer networks such as Usenet, Arpanet and CompuServe, from around the country and Canada contributed results for this table. The table would be thin and uninteresting without: Carol Parker, Lew Wolfgang, Larry Phillips, John Gilmore, Ali Ozer, Glenn Miller, Mike Howard, Gary Turton, Ian MacPhedran, Steve Neighorn, Steve Walton, Sandra Loosemore, Moshe Braner, Mike Berkley, Marion Hakanson, David Huenemoerder and Mitch Bunnell.

Roomers

by the Bandito

Despite the Commodore shakeups, the rumor mill continues with MORE tasty tid-bits.

The rumor mill almost stopped production while everyone took a deep breath after the shakeups within Commodore. They soon started stamping out new tidbits a week or so later.

More and more people are enjoying the Amiga and music, such as the players in the band Oingo Boingo. On the album cover of their latest, they thank Commodore for the Amiga! They also use the Amiga in their stage show. A Tom Petty video shows the Juggler in one sequence and an Amiga being sprayed with paint in another.

Word comes across the ocean of a company promising an Atari ST emulator for the Amiga. Who cares, right? The company is called Ahmed Research International by some, Ahmed International Research by others. I think the "AIR" acronym makes more sense. Mr. Ahmed claims the ST emulator will sell for 15 pounds, or about $25. You can call him at (011) 44-194-94442 to find out more. They also claim to have Macintosh and Commodore 64 emulators.

The Data Pacific "Magic Sac" Macintosh emulator is being readied for a late summer launch. It has a better chance of being a real product, especially considering it exists on the Atari ST. Don't get too excited about it. The ST version only runs about half the software on the commercial and public domain markets, according to a recent review in an ST magazine, but creator Dave Small said the Amiga version should run more software than the ST version.

Central Coast Software, makers of Dos-2-Dos and Disk-2-Disk, can reportedly read Macintosh disks with no additional hardware. Dos-2-Dos reads and writes MS-DOS and Atari ST formats; Disk-2-Disk reads and writes Commodore 64 disks. They hope to have a product like this out in early fall. Other reports say only four-fifths of the Mac format can be read at present and that there is much trouble calculating a 24-bit CRC checksum used on Mac disks. Hackers will always find a way, of course, so this may someday be possible.

Other rumors have surfaced that XenoSoft, the makers of the MS-DOS XenoCopy program, are preparing an Amiga version of their program that reads almost any 5 1/4 disk format, including Apple and CP/M formats.

Commodore shipped the first group of Sidecars in late May. The price is $995. So much for "significantly less than $1000," the long-time promise from Commodore. Also, Commodore is selling the PC-10-2, with two drives and a color monitor, for $995. Spring COMDEX in Atlanta should include an Amiga 2000 with a special board running a variant of Unix, along with a cost-reduced version of Commodore's PC clones.

The list price of the Amiga 2000 was raised to $1695, then late-breaking rumors were confirmed that the list price was raised to $1995. The Amiga 500 is now listing at $695. Some people believe this opens the way for a cost-reduced Amiga 1000 to fit in the price slot between the 2000 and the 500. The higher price for the 2000 also means it could be lowered some day, under the guise of a promotion.

Mass-marketing is still a strong possibility for the Amiga 500. Word has it that Sears and K-Mart have agreed to sell the machine. Sightings of warehouses of Amiga 500s have been reported in Canada and some people were saying two weeks until delivery at this time. Other sources say thousands of Amiga 500s have already been sold in Europe.

The best rumor of all said Commodore Germany has ordered 110,000 Amiga 500s, based on glowing sales figures from the first few weeks after its introduction. The rumor monger attached to this gossip said the Amiga 500 is selling much better than the Atari ST did when first introduced. Word has it that some Amiga developers danced, yelled and screamed when they heard of the order for 110,000 Amigas. It certainly means a new car for some developers and a strong future for our favorite computer.

The first production run of West Chester Amiga 2000s came off the production lines in mid-May, but these will, undoubtably, go to developers and dealers. A run of 3000 units is expected soon after that, in Germany or West Chester.

The new prices are as follows: The Amiga 500 model A501 512K memory expansion board at $199.95. Amiga 2000 peripheral model A2002 RGB analog monitor at $399.95. A2010 3 1/2 inch internal drive at $199.95. A2052 2 megabyte RAM expansion at $499.95. A2088 BridgeCard at $499.95. A2090 hard disk controller at $399.95. Note that the medium-persistence phosphor A2002 monitor sells for the same price as the current Amiga monitor and that an Amiga 500 with monitor costs the same as a straight Amiga 500.

continue...

The trade-up deal from the Amiga 1000 to the Amiga 2000 sounds like a dead issue at this time, but the dealer meeting at COMDEX should reveal more of Commodore's plans.

A visitor to the new Amiga support office that replaced the Los Gatos reports that they are working on a process to re-compile the Amiga operating system on an Amiga 2000. They are working to use the Manx compiler on a native Amiga, instead of the Greenhills compiler on a Sun. No word yet on comparative code size. Jay Miner is back at work consulting, six long months after Commodore "gave him an offer her couldn't refuse" to end his 5 year contract. He'll be working on pacemaker chips for a company called Ventritex in Sunnyvale and for a toy company.

Electronic Arts reports Earl Weaver Baseball went into final mastering in late May and should ship at the end of June. Insiders also report RJ Mical's game may never see the light of day because EA officials are upset that he's working on other projects now.

Other products from EA this summer should include Ferrari Formula One, the race-car game mentioned in this column a long time ago, a Deluxe Music data disk called "Hot & Cool Jazz" and a typing tutor called "Intellitype." There is quite a history behind Intellitype. EA product development had originally discussed an adult-oriented typing tutor, to differentiate it from all the other typing tutors out there. It had R-rated typing exercises, mostly racy stories to re-type, but they changed their minds at the last moment. Other titles from the EA affiliated labels such as Origin and Sierra should be forthcoming.

Rumor scouts report work is being done on an Amiga-controlled laser toaster. They think they can sell a lot of these to ritzy hotels for making custom designs on breakfast toast. The laser draws the hotel's logo on the toast or can accept IFF files for imprinting in sixteen shades of brown. Developers are not sure if the device will be on the serial or the parallel port.

The people who make the PCLO circuit design software, SoftCircuits, are developing slow scan television software for the Amiga. This variant of ham radio sends television pictures over the airwaves. Someone relays that MIT Wisdom Prolog for the Amiga is targeted for a July release.

•AC•

ALL ABOUT PRINTER DRIVERS

by Richard Bielak

Introduction

The day my Amiga came home, I set it up next to my trusty C.Itoh Prowriter printer. I was able to connect the printer and computer, but I wasn't able to print any of those great Amiga graphics— the Prowriter was not one of the choices in the Preferences!!!

At that time (nearly a year ago) little information was available on making the Amiga work with "strange" printers. I mustered my courage, rummaged through the developer's kit and pulled out the "ROM Kernel" manual. I turned to the "Printer Device" chapter and started reading...and a few weeks later I was able to print Deluxe Paint pictures on the Prowriter!

In this article, I describe what I have learned through my labors. I hope this information helps you with printer problems.

Devices and Device Drivers

Before jumping into the technical details, let's consider some basic ideas. A computer system consists of a CPU, memory and external devices. These devices are prompted for either input, output or both. For example, a printer is used for output only, but a modem is used for both input and output.

Every device connected to the computer is controlled in a unique way. Therefore, distinct programs must control each device. On the other side of the fence, an application program needs a uniform interface to all devices. This uniformity makes programming easier, as the programmer must learn only one method of handling I/O, instead of worrying about device specifics.

The operating system provides the uniformity of interface between programs and devices through the following scheme. An application program submits a standard I/O request to the system. The system performs intial processing and then dispatches the request to the device-dependent code. This device-dependent code tells the device what to do.

The Amiga DOS Printer device

Amiga DOS isolates the user and application programs from the details of printing using the PRT: device. To produce printer output, a program simply writes to PRT:.

Using the PRT: device for printing has many advantages. First, you don't have to know which port— either serial or parallel—is connected to the printer. Second, only one set of control sequences chooses printer features. Finally, a single method prints graphics.

So, with many different printers out there, how can this be done? The answer is simple. The PRT: device driver makes all this "magic" possible. It knows where to send the data, which control sequences are specific to your printer and how to print graphics.

To better understand how the PRT: works, let's consider what happens when a text file is printed. To send a file to the printer, we issue a simple CLI command, like "COPY FILE.TXT PRT:". First, Amiga DOS makes sure that the PRT: device driver is in memory. If not, the driver is loaded from the Workbench disk (from the directory "devs"). Next, the initialization code of PRT: driver is called. This code sets up the printer according to the Preferences.

After all this preparation, the data from the file is given to the PRT: driver. The driver, in turn, sends the data to the printer, while also looking for control sequences. A control sequence begins with the ESC (27 ASCII) character. The driver translates each control sequence into a printer specific sequence. Note that if a control sequence is not defined for your printer, it is deleted from the data stream.

Now that we have a better idea how the PRT: device works, let's see how to configure the Amiga to work with a specific printer.

User's View

The configuration of the PRT: device is done through Amiga Preferences. Two Preference settings are most important in getting the printer to work. One specifies which port the printer is connected to. The second names the particular printer type.

continued...

If your printer is connected via the parallel interface, selecting parallel port in the Preferences is enough to enable printing. For a printer using the serial interface, in addition to selecting the serial port, you must set the speed of data transmission. Make sure the speed set on the Amiga matches the speed set on the printer.

In order to use features activated via control sequences (usually including graphics), you must select the appropriate printer type in the Preferences. For example, if you own an Epson JX-80, then select "Epson_jx-80" as your printer type.

Specifying printer type allows the PRT: driver to use printer specific code when translating control codes or printing graphics. The printer specific code is stored in load files in the "devs/printers" directory on the Workbench disk.

If you are lucky enough to have a printer named in Preferences, just select the port, pick the printer type and you can start printing. If you own a "strange" printer (such as C. Itoh Prowriter), you're out of luck! Well, not quite. There are still a number of things you can do to get printout.

One method of printing is using the "generic" printer driver. You can choose this driver by selecting "Custom" as the printer type in Preferences and "generic" for the custom printer name. The "generic" driver strips all control sequences and allows you to print text. Unfortunately, you cannot print graphics with the "generic" driver.

Another trick lets you print text and use some features turned on by control sequences (such as underline, bold, etc). You can imbed printer specific codes into the text before printing. Be careful, though. Since the PRT: driver removes control sequences, we must bypass the PRT: completely. Bypassing the PRT involves sending the text directly to the SER: or PAR: device (depending on which port your printer is connected to).

There's still a problem, though. This trick cannot print any graphics! To print graphics, you must make the driver specific to your printer (provided your printer can produce graphics in the first place). Before trying to write your own printer driver, check the Public Domain. Your work may have already been done! (See the end of this article for a list of sources for PD printer drivers).

Once you find the appropriate printer driver, you have to install it on your system. Installation is a simple, two-step procedure. First, put the printer driver file in the "devs/printers" directory on the Workbench disk. Next, go into the Preference's printer set up. Select "Custom" as your printer type and enter the name of the printer driver file in the space labeled "Custom Printer Name" (this field usually contains "generic"). You should be able to print graphics and text!

Some common printer problems

Let's consider some common printing problems and possible solutions. I encountered some of these problems myself; my friends ran into others.

Let's start with a big problem— suppose you can't print anything! Although this problem is usually not caused by the printer driver, you should still check the Preferences to make sure you've chosen the correct port. If you are using the serial interface, make sure the printer's transmission speed and the serial port are the same. Try sending a test file to either the PAR: or the SER: device to bypass PRT: driver. If none of the above solves the problem, either the printer or printer cable is broken.

Another common problem is the inability to print graphics. In this case, the printer driver is usually at fault, but other components should still be checked. Not all printers can print graphics. For example, most daisy wheel printers cannot print fancy graphics. If your printer can produce graphics, and you have the appropriate printer driver, but still cannot print pictures, something else must be wrong. The problem could be your printer cable.

If you are using the parallel interface, the data is sent to the printer through eight separate wires. When printing, only seven of those wires are used, as all ASCII characters are represented by seven bits. Graphic data is sent in eight bit bytes. Therefore, all eight wires are needed. If your cable is missing the eighth wire, you cannot print pictures, even though text will print with no trouble.

Another source of many strange problems is printer incompatibility. A printer claiming to be Epson compatible, might be only 90 percent compatible. Such a problem usually results in strange characters appearing unexpectedly in your documents. Garbage characters print, even if you do **not** use control sequences!

Here's a brief explanation of how this happens. The PRT: driver initializes the printer each time a file is printed by writing the appropriate control sequence to the printer. If you are using the Epson driver, but your printer is not 100 percent Epson compatible, parts of the initial control sequence can appear as "garbage" on top of your document. Unfortunately, the only sure solution to this problem is to use a printer driver written for that particular printer.

When dealing with printing problems, it is easy to determine if the printer driver is the cause. Just bypass the printer driver by sending files directly to the PAR: or the SER: device.

Programmer's view

If you can't find the appropriate printer driver on Public Domain disks and Bulletin Boards, just one option remains. You will have to write one!

To write a printer driver, you'll need Lattice C compiler, the 68000 assembler and "ROM Kernel" manual. As far as I know, printer drivers can only be written using Lattice C and must be linked with ALINK.

To be more precise, you need a "printer dependent code segment." The printer dependent code segment consists of some tables and subroutines used by the PRT: device driver.

From this point on, for simplicity's sake, "printer driver" will actually mean "printer dependent code segment."

Many source files (assembler and C) make up a printer driver. Some of these files may be missing from the developer's kit. For instance, there was no source for a sample printer driver. Luckily, all these files are listed in the "ROM Kernel Manual" and can be typed in! To avoid all this work, you should try to get the source of a Public Domain printer driver. Usually, all the needed source files are bundled in with the driver.

Creating a new printer driver involves modifying four files: PRINTERTAG.ASM, DATA.C, DOSPECIAL.C and RENDER.C. PRINTERTAG.ASM and DATA.C contain tables. DOSPECIAL.C and RENDER.C contain subroutines used by the PRT: driver. DOSPECIAL.C takes care of text related work. RENDER.C is used for graphics.

The file PRINTERTAG.ASM holds the **Printer Extended Data** table (or "PEDData" table). "PEDData" describes various printer characteristics and functions that perform all printer operations.

Two fields in "PEDData" table define the type of printer: **printer class** and **color class.** Printer class tells the driver if the printer is capable of producing graphics and whether the graphics are color or black and white. Color class specifies the printer's method of printing colors. For example, the Epson JX-80 printer produces all colors by mixing four basic colors: yellow, blue, magenta and black.

Other entries in the PED table provide information for printing graphics, including: the dot density per inch (both vertical and horizontal), maximum number of dots (horizontal and vertical) and the number of rows printed during a single pass of the print head. The ratio between the number of horizontal and vertical dots per inch determines the aspect ratio of the printed picture. The best way to choose these values is to pick the maximum horizontal value and determine the correct count for the vertical direction through trial and error.

The file DATA.C contains a table used by the PRT: driver to translate the standard Amiga control sequences into printer specific sequences. For each Amiga sequence, there is a corresponding entry with the printer control sequence. When no direct translation exists, the entry in the table is a single byte, containing the value 255. For any standard sequence that "translates" to this byte, the PRT: driver calls a subroutine to perform this function (provided the printer can support it). This subroutine is called "DoSpecial" and is contained in the file DOSPECIAL.C.

DoSpecial's parameters include a function to perform and a buffer. If "DoSpecial" can handle the function, it fills the buffer with the correct control sequence. The data in the buffer is written to the printer after "DoSpecial" returns. For example, every "DoSpecial" subroutine has code to handle printer initialization. In this case, a simple translation of control sequences will not do. . . the printer must be set up according to the Preferences.

The final, most complicated portion of the printer driver is the "Render" funtion (contained in the file RENDER.C). As the name suggests, "Render" is used for rendering graphics. It's important to understand how "Render" works, especially when solving printer problems.

"Render" works with only a small portion of the picture at a time——the data "Render" holds is just enough for a strip as wide as the print head. If printing is black and white only, then "Render" keeps all the data in one buffer. Color printers usually produce one color in a single pass of the print head, meaning that printing one strip of a color picture requires a pass for each color. Therefore, for a color printer, "Render" must hold a data buffer separately for each color.

The buffer held by "Render" represents a bit map of a strip of the picture. The buffer is set up so the graphics are produced when the buffers content is sent to the printer. For printers that allow dot addressable graphics, this buffer contains a control sequence introducing graphics data, followed by the actual data (terminated by a carriage return and line-feed). For example, for a black and white Epson printer, the buffer holds data representing strips eight dots high and 960 dots wide. A color Epson needs four such buffers, one for each color. In either case, one byte represents eight vertical pixels. Some of the newer dot-matrix printers use 24 vertical pins. For a 24-pin printer, three consecutive bytes in the buffer represent 24 vertical dots.

"Render" is called by the PRT: driver with four parameters: status byte, color number and X and Y coordinates. The status byte specifies what is being done. Possible actions include: master intialization, adding a pixel to a buffer, writing data to the printer and cleaning up.

First, "Render" is called to perform master intialization. This process includes setting up the printer for the appropriate graphics mode, allocating memory for data buffers and initializing all variables.

Render's second function places pixels in the buffer — more precisely, turning on bits in the appropriate byte in the buffers. The "x" and "y" parameters determine which bit will be turned on. The color parameter determines which data buffer the pixel goes into.

Once the data buffer is filed with pixels, "Render" writes the buffer to the printer. The write is accomplished via a call to a lower level I/O function. The printer data block, another structure maintained by PRT: driver, contains pointers to functions that do the actual I/O. The advantage of using these functions is that "Render" doesn't know anything about the gory details of I/O (such as which port is used by the printer).

Finally, when the entire picture is printed, "Render" is called to clean up. Buffers must be deallocated and the printer must be reset.

continued...

As you can see, the reason printing of graphics is so slow is the need to call "Render" for nearly every pixel in the picture. In low res mode at least 64000 calls (i.e. 320 times 200) are needed! It gets worse for color printers, where a separate call for each color must be made. One way to speed up the printing process is to use double buffering. While one buffer is printing, the other can be filling with pixels.

Pitfalls

Debugging printer drivers is a difficult task. Since a printer driver is actually just a collection of routines called from the PRT: device driver, many common debugging methods cannot be used. For example, inserting "printf" into the code does not work! Not only is the driver linked without the standard I/O routines, but also none of the necessary libraries are opened for the driver task. In addition, the PRT: device driver doesn't have a window for "printf".

One possible solution is to write a custom "printf" function that works from within the driver. Other approaches include using the ROM Wack debugger or the "low-tech" debugging method— reading the code!

During debugging, viewing the control sequences sent to the printer by the driver can be very helpful. Some printers can be set to a mode so any received data is dumped in hex. If you can see exactly what data was sent to the printer, your task may be a little easier.

Printer drivers linked with the current version of ALINK do not include most of the Lattice runtime support routines. Therefore, your C code generates calls to functions that are **NOT** linked to the driver. A usual symptom of this problem is an unresolved linker reference to a strange looking symbol (something like "CX4123"). If you try to use a driver linked with such errors, you'll probably crash the Amiga when printing.

I've run into this problem while using muliplication, division and modulo operators on LONG integers. Apparently Lattice C uses runtime routines for these computations. Since I'm not a C compiler expert, I changed my code to avoid using these operations.

Summary

Although dealing with printers and printer drivers can be a frustrating experience, remember why the drivers are there in the first place——thanks to printer drivers, almost any printer can be used with the Amiga.

The designers of the Amiga DOS did a good job isolating the user and application programs from the gory details of printing. After all, that's how it should be!

Where to find printer drivers

Many printer drivers can be found in Public Domain. *AC* lists drivers available on the AMICUS disks. Here is a list of some drivers I noticed on CompuServe:

- Epson - enhanced Epson driver (some bugs fixed).
- NEC P6 - nice driver for the 24-pin printer.
- LA-50 - driver for the DEC LA-50 printer.
- SG-10 - SG-10 Star, driver by Star Micronics.
- CANNON - Cannon Inkjet PJ-1080A.
- CGP-220 - Radio Shack CGP-220 Inkjet; supports graphics.
- OKI 292 - text and graphics.
- GEM-10 - Gemini-10; supports graphics.
- MTSP 80 - Tally Spirit 80; supports graphics.
- C.Itoh - text and graphics (by yours truly).

The New York Amiga user's group (AMUSE) also distributes a PD disk called "Joy of Printing." This disk can be obtained for $5 at:

AMUSE
New York Amiga Users
151 1st Ave, BOX 182
N.Y.C., N.Y. 10003

•AC•

Explore the Potential of Amiga!

NEW YORK	LOS ANGELES	CHICAGO
October 10-12, 1987	January 22-24, 1988	July 22-24, 1988

Featuring

Keynote Sessions

Jay Miner, the Father of the Amiga™, will open the New York **AmiEXPO.** ***R. J. Mical,*** the Designer of Intuition, will provide insights into software development.

Exhibition Hall

A sampling of exhibitors:

Activision, Inc.
Amazing Computing
Amigo Business Computers
ASDG, Inc.
Brown-Wagh Publishing
Byte by Byte
Central Coast Software
Computer Systems Associates
Creative Microsystems, Inc.
Firebird Licensees, Inc.
Lattice, Inc.
Liquid Light, Inc.
Manx Software Systems
Microillusions
NewTek, Inc.
New Horizons Software
PiM Publications, Inc.
Octree Software
Word Perfect Corporation

Amiga™ is a registered trademark of Commodore-Amiga, Inc.

Development Forums

Intensive working sessions with the leading Amiga developers, such as *NewTek,* detailing specific products.

User Seminars

- The Architect's Amiga
- Art Direction and the Amiga
- The Amiga in Video Production
- Amiga MIDI: Lights, Sound, Action!
- The Ultimate Game Machine, Amiga
- Amiga's Desktop Color Publishing
- "Vax in a Box" - Amiga Engineering

* AND MORE *

For information call ***800-32-AMIGA*** *(in New York call 212-867-4663) or complete the form and return it to:*

AmiEXPO Headquarters

YES! Send me more info on **AmiEXPO!** AC6

Name _____

Company _____

Address _____

_____

_____

Telephone _____

Return form to:
AmiEXPO Headquarters
211 East 43rd Street, Suite 301
New York, New York 10017

INTUITION GADGETS

Proportional Gadgets...

by Harriet Maybeck Tolly

Welcome to the third installment of 'Intuition Gadgets.' Gadgets provide a user interface for your application. Three types of Gadgets are available:

- String Gadgets (described in Volume 2, Number 3 of Amazing Computing),which accept strings of ASCII text.
- Integer Gadgets, a type of String Gadget that accepts only integers.
- Boolean Gadgets (described in Volume 2, Number 5 of Amazing Computing), hit and toggle types.
- Proportional Gadgets, or sliders, described this month.

This series is not meant as a substitute for the Intuition manual. You'll want a copy of the Intuition manual or one of the other programming books available for basics, such as associated structures and flags.

As with previous installments, I'll be covering areas which can be confusing, have changed in AmigaDOS V1.2 or that still contain bugs. I continue to stress that these problem areas are only a small part of Intuition as a whole. Gadgets are an easy-to-program interface, appropriate for a broad range of applications.

This month we'll be finishing up our journey through Gadget-land with a look at Proportional Gadgets.

Proportional (Prop) Gadgets are most often implemented as sliders, which allow the user to scroll through a list of information. Prop Gadgets can have movement in both the horizontal and vertical direction, even at the same time. It is easy to imagine an interesting 'game-type' control that could be implemented this way.

Prop Gadgets seem to have the fewest bugs of gadgets in general. On the other hand, many programmers find the values associated with Prop Gadgets confusing to orchestrate.

A Prop Gadget has as its SpecialInfo member, a pointer to a PropInfo structure. This structure is defined as follows:

```
struct   PropInfo
{
USHORT   Flags;
USHORT   HorizPot; /* Horizontal quantity percentage */
USHORT   VertPot;  /* Verticle quantity percentage */
USHORT   HorizBody;/* Percentage of body shown at once */
USHORT   VertBody;
USHORT   CWidth;   /* Container real width */
USHORT   CHeight;  /* Container real height*/
USHORT   HPotRes, VPotRes; /* Pot increments */
USHORT   LeftBorder; /* Container real left border */
USHORT   TopBorder;  /* Container real right border */
};
```

A Prop Gadget is composed of two parts. The piece that the user drags with the mouse is the knob. The area in which the knob can move is the container. Programmers can supply a custom image for the knob and/or a border for the container. For example, a border can be used to make a Prop Gadget appear as an oval. The actual container, however, will still be a rectangle. If you can live with a rectangular knob in a rectangular container, set the AUTOKNOB flag. This will cause Intuition to supply the whole image for you. You must declare an Image structure for Intuition to use, but you do not need to initialize it.

The following discussion will use the most common example of Prop Gadget use - as a slider to scroll through file names. I'll be using the following variables.

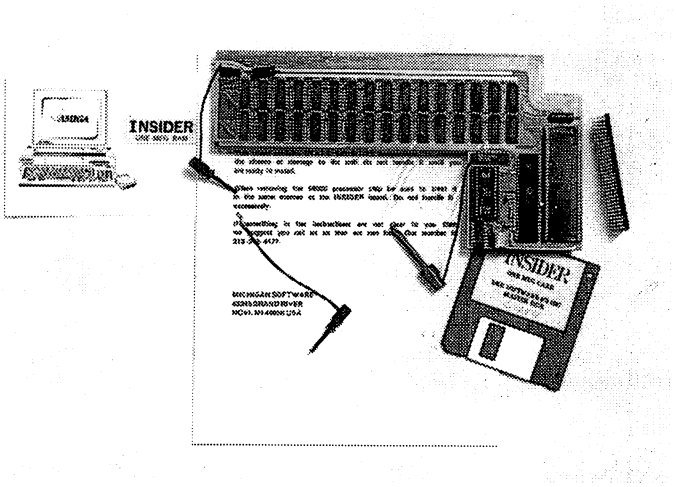
total_names	-	Total file names available to the user.
first_name	-	The index (starting at 0) of the first name currently displayed in the list.
visible_names	-	Number of names visible to user at any one time.

Body and Pot values

Your program is responsible for setting the value of the Body. This is the size of the knob. If your knob moves vertically, you will want to initialize the VertBody member and set the FREEVERT Flag. If your knob moves horizontally, initialize the HorizBody member and set the FREEHORIZ Flag.

Note that although FREEVERT and FREEHORIZ allow and/or restrict motion of the knob within the container, the coordinates of the pointer (reported to you in IntuiMessage.MouseX and IntuiMessage.MouseY) will continue to have motion in both directions. In other words, if only FREEVERT is set, the knob will only move vertically. Mouse coordinates, however, will continue to change both vertically and horizontally.

continued...



THE INSIDER
from Michigan Software

The 'FIRST' plug in no solder, internal memory expansion board for the Amiga.

Adds one full meg of Memory to your Amiga 1000 so if you have 512K now you get 1.5 meg total. Can also be used on a 256K machine giving you 1.1 meg.

* REALTIME CLOCK/ CALENDAR
* AUTO CONFIG UNDER 1.2
* FAST MEMORY
* WORKS WITH SIDECAR
* ONE YEAR WARRANTY

For those people who have external memory boards and need to go one meg further the INSIDER is the answer. Fully compatible with external boards.

O N L Y $ 349.95
shipping 3.00

VHS / BETA DEMO TAPE AVAILABLE SHOWING INSTALLATION & INFORMATION
* $10.00 / RETURNED TAPE REFUNDABLE TOWARDS PURCHASE OF INSIDER

ORDERS AND INFORMATION:

PHONE ORDERS (313) 348-4477

AMIGA BBS (313) 348-4479

OR CALL FOR A DEALER NEAREST YOU

MICHIGAN SOFTWARE DIST INC

43345 GRAND RIVER AVE

NOVI MICHIGAN 48050

VISA / MC / AMEXP / C.O.D.

* KWIKSTART

(1.2 KICKSTART IN ROM, ALLOWS FASTER STARTUP TIME, BUT DOESN'T LOCK YOU INTO 1.2, SWITCHABLE FEATURE LETS YOU STILL USE DISK BASED KICKSTART, PRESS CONTROL AMIGA/AMIGA FOR 3 SECONDS AUTOMATICALLY SWITCHES FOR YOU. PLUGS INTO THE 68000 PROCESSOR AND REQUIRES ONE PAL CHANGE ON THE DAUGHTER BOARD, LESS WORK AND MORE FEATURES THAN OTHER 1.2 KICKSTART KITS. IT'S ALSO COMPATIBLE WITH THE INSIDER MEMORY BOARD.

RETAIL $ 169.95

The accepted interpretation of the size of the body is as follows:

Body = Percent of total items visible at one time.

Body values range from (hex) 0x0001 (smallest) to 0xFFFF (largest). So, if you can see all the items (100%), your knob should fill the whole container, or be equal to 0xFFFF. If you are currently viewing half the total names, your Body would be 0x7FFF. The general formula is:

```
Body = (ULONG)(visible_names * 0xFFFF) / total_names;
```

The Body value will determine the amount the knob moves when the user clicks above/below/right/left of it, but not on it. If the Body is 0x7FFF, it fills half the container. For a Gadget with verticle motion, clicking below the knob will cause the knob to jump down to the bottom half of the container. In other words, it moves in 50% incerments. This seems obvious with an AUTOKNOB image. But imagine that you have specified your own image that is only 1/10 of the container. You can still specify a Body of 0x7FFF. When you click below the image in the container, the knob will jump all the way down to the bottom, even though it is not half the size of the container.

The Pot (derived from potentiometer) member of the PropInfo structure also ranges from 0x0000 to 0xFFFF. When the knob is all the way to the left/top, the Pot will be 0x0000. When the knob is all the way to the right/bottom, the Pot will be 0xFFFF. There is no way to reverse the scale in Intuition, although it is simple enough to do this translation with the formula:

```
Translated_Value = 0xFFFF - Pot;
```

Most often, the Pot of the knob is initialized to 0x0000. However, in some cases, you may want to set the Pot of the knob to some other value. If you know the index of the first file name you wish to display, use the following formula to determine the Pot:

```
Pot = MIN (0xFFFF,(first_name <<
                16 / total_names - visible_names));
```

Interpreting Pot values

Finally, you'll want to read the Pot value of the Prop Gadget, after you have determined that the user has altered it from its initial value. Intuitively, we can see that a Pot value of 0x0000, should mean that we display the top of our list. Conversely, if our Pot is 0xFFFF, the bottom of the list is displayed.

The original Intuition manual gave a very complicated example for interpreting Pot values. I'll be brave and admit that I'm not sure of the application for this time-related example. Instead, let's continue with our list-of-file-names example.

When the user alters the Pot value by sliding the knob up or down, you'll want to determine the index of the new first_name. One possible formula is:

```
first_name = (Pot * (ULONG)
        (total_names - visible_names) + (1<<15)) >> 16;
```

These formulas are basically the same as those found in the Enhancer manual. If anyone has read the explanation in the Enhancer manual and wondered why the number 75 suddenly jumps into the picture, it's because the example is for a specific case. However, somehow in the printing of the manual, the explanation of this case was omitted. It should be:

```
total_items = 100
visible_lines = 25 (one-fourth of 100)
```

This explains why a Pot of 0xFFFF would produce a first_line of 75.

You will see other formulas for determining first_name. They are all similar. Some have less chance of overflowing, while some execute faster. The one above, from the Enhancer Manual, seems to be the sanest option.

Knob movement

Now, let's look at how we might determine that the user has altered the Pot of the knob. The simplest way is to set the RELVERIFY flag of the Gadget. This will cause your program to get a GADGETUP IDCMP message when the user releases the left mouse button. At this point, you can check the Pot value and adjust the first_name accordingly.

Although the Intuition manual states that RELVERIFY triggers a GADGETUP message only if the pointer is still over the gadget, this is NOT the case for Prop Gadgets. Once the user has clicked the select button over the gadget, he can then move the pointer off the slider, keeping the select button down. The knob will continue to move back and forth in the container while the pointer is moved and you will get a GADGETUP message when the select button is released, regardless of whether it is still over the gadget.

Another method used by some programmers is to update the list while the user is still sliding the knob around, before the left mouse button is released. This is a little trickier. First, we must set the FOLLOWMOUSE flag for the Gadget, in addition to the GADGIMMEDIATE and RELVERIFY flags. Setting the FOLLOWMOUSE flag will cause MOUSEMOVE IDCMP messages to be sent to your program. You'll get the GADGETDOWN message when the user clicks the select button down over the Gadget. Then, you'll get a stream of MOUSEMOVE messages as the user slides the knob around. You should wait until there are no messages waiting at the UserPort of the Window. Then, using the most current Gadget values, you can check the Pot and update the first_name accordingly. Finally, you'll get the GADGETUP message when the select button is released.

If you do not set GADGIMMEDIATE you will still receive MOUSEMOVE messages as the mouse is moved. You will also receive a MOUSEMOVE message when the user clicks on the gadget, at the point where you would have received a GADGETDOWN message. You get this even if there is no mouse movement on the click.

The example program below gives an example of how to use MOUSEMOVE messages to track the Pot value.

Fuzzies

OK, now for a pop quiz. Describe a fuzzy. (Those of you not following from previous gadget articles are excused). The answer is... bits of ghosting that appear where they should not. With String Gadgets, we saw fuzzies when the width of our gadget wasn't a multiple of the width of the font we were printing into it. Prop gadgets have a version of fuzzies all their own. Trust me, this one is weird.

First, the detail and block pens need to be set to anything other than the default colors of the screen. On Workbench, this is color 0 for detail and color 1 for block. Set the pens to something other than 0,1 or -1,-1 (which defaults to the screen's colors). Now, if in your list of gadgets, a Prop gadget points to a disabled String gadget, the Prop gadget will have a 'fuzzy' container. It will operate fine, but its container will have the ghosting pattern. Simply avoiding having a Prop gadget point to a disabled String Gadget in your list will eliminate this problem. The other solution is to live with the default screen colors.

The example program below includes a Prop Gadget pointing to a String Gadget for you disbelievers.

Border Gadgets

Putting your gadgets in the borders of your windows sounds desirable, but can be confusing to implement. The Intuition manual says you can 'tuck your own window gadgets out of the way into the window border.' I took this to mean that clipping would be done at the inner edge of the gadget, just as it would be done at the edge of a normal window. This is not the case. If, for example, you write IntuiText into your window, it will not be clipped at a border gadget. There is one piece of information that can make the clipping that you will need to do a little easier.

Regardless of whether your have specified a GimmeZeroZero window, the value Window.GZZWidth and Window.GZZHeight *will* contain the actual, correct interior width and height values, taking your gadgets into account. This means you can clip against these values, instead of having to add or subtract your gadget width from the window size to calculate the value to which to clip.

continued...

Let's look at an example. Suppose we put our Prop Gadget in the right border of the window, with a column of IntuiText to the left of the Prop Gadget. We can now clip the Text against the Window.GZZWidth value.

Verion 1.2 AmigaDOS Enhancements

A new version of ModifyProp(), appropriately enough, called NewModifyProp() was added for V1.2. The old version called RefreshGadgets(). As I have mentioned in previous articles, this routine causes excessive flashing by re-rendering all gadgets in the list. NewModifyProp() has an added parameter which lets you specify the number of gadgets to refresh.

You can now specify GADGHIMAGE to tell Intuition that you have an alternate image for highlighting your knob. The Enhancer manual suggests that this image should be the same size as the initial image. I experimented with pointing to an alternate knob image larger than my primary knob image and vice versa. In both cases, the alternate image was rendered very prettily, but very wrong.

Listing One is a sample 'C' program which opens a window with Prop Gadgets on it.

- Prop Gadget used to scroll through list of strings.
- Prop Gadget points to a disabled String Gadget and, therefore, has fuzzies.
- Prop Gadget has alternate image for highlighting the knob.
- Prop Gadget has motion freed in both directions and is borderless.

About the author

Harriet Maybeck Tolly owns a software company in Wilmington, Massachusetts called TollySoft. She and her husband Bob are working on Amiga software. She can be reached at:

PLink - TollySoft
BIX - rtolly
Usenet - rtolly@CCA.CCA.COM
GEnie - TollySoft

Listing One

```
/*************************************************************
  'C' program showing examples of Proportional Gadgets.

  This was compiled using Manx, Aztec "C", AmigaDos V1.2
  The code depends on V1.2 functions to operate correctly.
  It is intended to be run from CLI.

  cc prop
  ln +cdb prop.o -lc

  Copyright (C) 1987 H. Maybeck Tolly, TollySoft

  This program is in the public domain and may be
  distributed free of charge.

*************************************************************/

#include <exec/types.h>
#include <exec/exec.h>
#include <intuition/intuition.h>
#include <graphics/gfxbase.h>
#include <functions.h>

#define visible_lines 5L
#define total_lines 10L
#define MAXLEN 20

struct IntuitionBase *IntuitionBase = 0L;
struct GfxBase *GfxBase = 0L;
struct Window *ControlWindow = 0L;
struct IntuiMessage *MyIntuiMessage;


/***********************************/
/* Declare Prop Gadget structures. */
/***********************************/

/* This Prop Gadget shows how to read Pot values to */
/* scroll through a list of strings. */

struct Image g_image_scroll;
struct PropInfo g_prop_scroll;

struct Gadget PropGadget_scroll = {

   NULL,              /* pointer to Next Gadget */
   180, 30, 20, 60,   /* (Left Top Width Height) Hit Box */
   GADGHCOMP ,        /* Flags */
                      /* Activation flags */
   GADGIMMEDIATE | FOLLOWMOUSE | RELVERIFY,
   PROPGADGET ,       /* Type */
   (APTR)&g_image_scroll,   /* pointer to Image */
   NULL,              /* no pointer to SelectRender */
   NULL,              /* pointer to GadgetText */
   0,                 /* MutualExclude  not implemented*/
   (APTR)&g_prop_scroll,    /* pointer to SpecialInfo */
   0,                 /* no ID */
   NULL               /* no pointer to special data */
};

/* This Prop Gadget points to a disabled String Gadget,  */
/* to show a "Prop Fuzzy".                               */

/* This String Gadget is used only to show a property    */
/* of Prop Gadgets. It is not intended as a clear example*/
/* of a string gadget. Please refer to previous articles */
/* in this series for more info on String Gadgets.       */

struct StringInfo StrInfo = {
         (UBYTE *)"static test string ",
         NULL, 0, 20, 0,0,0,0,0,0,NULL,0,NULL};

struct Gadget StrGadget = {

   &PropGadget_scroll,
   220,30,160,7,
   GADGHCOMP | GADGDISABLED ,
   GADGIMMEDIATE | RELVERIFY,
   STRGADGET,
   NULL, NULL, NULL, 0, (APTR)&StrInfo, 0, NULL
};

struct Image g_image_fuzzy;
struct PropInfo g_prop_fuzzy;

struct Gadget PropGadget_fuzzy = {

   &StrGadget,           /* pointer to Next Gadget */
   400, 30, 20, 60,  /* (Left Top Width Height) Hit Box */
   GADGHCOMP ,           /* Flags */
   GADGIMMEDIATE | RELVERIFY,       /* Activation flags */
   PROPGADGET ,          /* Type */
   (APTR)&g_image_fuzzy, /* pointer to Image */
   NULL,                 /* no pointer to SelectRender */
   NULL,                 /* pointer to GadgetText */
```

continued...

```
    0,                  /* MutualExclude not implemented*/
    (APTR)&g_prop_fuzzy,  /* pointer to SpecialInfo */
    0,                  /* no ID */
    NULL                /* no pointer to special data */
};


/*******************************************************/
/* All images  must reside in chip memory (lower 512K). */
/* If your compiler/linker does not have an option to do*/
/* this, you must allocate chip memory and copy this   */
/* data into that memory before assigning the pointer   */
/* members of the Gadget structure.                */
/*******************************************************/

UWORD prime[] =
    {
/* plane 1 */
    0x0200,
    0x0700,
    0x0700,
    0x0F80,
    0x0F80,
    0x1FC0,
    0x1FC0,
    0x3FE0,
    0x7FF0,
    0x3FE0,
    0x1FC0,
    0x1FC0,
    0x0F80,
    0x0F80,
    0x0700,
    0x0700,
    0x0200,

/* plane 2 */
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF
};


UWORD alt[] =
    {
/* plane 1 */
    0x7038,
    0x3870,
    0x1CE0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x0FC0,
    0x1CE0,
    0x3870,
    0x7038,

/* plane 2 */
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,

    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF,
    0xFFFF
};

struct Image g_image_primary = {
 1,0,13,17,2,
 prime,
 0x03, 0x0, NULL
};

struct Image g_image_alternate = {
 1,0,13,17,2,
 alt,
 0x03, 0x0, NULL
};

struct PropInfo g_prop_image;

struct Gadget PropGadget_image = {

    &PropGadget_fuzzy,       /* pointer to Next Gadget */
    500, 30, 21, 60,  /* (Left Top Width Height) Hit Box */
    GADGIMAGE | GADGHIMAGE,  /* Flags */
    GADGIMMEDIATE | RELVERIFY, /* Activation flags */
    PROPGADGET ,             /* Type */
    (APTR)&g_image_primary,  /* pointer to Image */
    (APTR)&g_image_alternate,/* no pointer to SelectRender*/
    NULL,                    /* pointer to GadgetText */
    0,                       /* Mut-Exc not implemented */
    (APTR)&g_prop_image,     /* pointer to SpecialInfo */
    0,                       /* no ID */
    NULL                     /* no special data */
};

struct Image g_image_free = {
 1,0,13,17,2,
 alt,
 0x03, 0x0, NULL
};

struct PropInfo g_prop_free;

struct Gadget PropGadget_free = {

    &PropGadget_image,     /* pointer to Next Gadget */
    200, 120, 80, 35,  /* (Left Top Width Height) Hit Box */
    GADGHNONE | GADGIMAGE, /* Flags */
    GADGIMMEDIATE | RELVERIFY, /* Activation flags */
    PROPGADGET ,           /* Type */
    (APTR)&g_image_free,   /* pointer to Image */
    NULL,                  /* no SelectRender */
    NULL,                  /* pointer to GadgetText */
    0,                     /* Mut-Excl not implemented */
    (APTR)&g_prop_free,    /* pointer to SpecialInfo */
    0,                     /* no ID */
    NULL                   /* no pointer to special data */
};

/*******************************************************/
/* IntuiText structures for the list of items to scroll */
/*******************************************************/
```

```
   struct IntuiText IntuiStr[5] = {
         {2,3,JAM2,5,30,NULL,NULL,NULL},
         {2,3,JAM2,5,45,NULL,NULL,&IntuiStr[0]},
         {2,3,JAM2,5,60,NULL,NULL,&IntuiStr[1]},
         {2,3,JAM2,5,75,NULL,NULL,&IntuiStr[2]},
         {2,3,JAM2,5,90,NULL,NULL,&IntuiStr[3]}
};

/*******************************************************/
/* Text strings through which to scroll.               */
/* In our example they will retain the same values     */
/* throughout the program. In a real application they*/
/* would most likely change. We won't pad them with    */
/* blanks now, so that we can show how to do it on     */
/* the fly.
*/
/*******************************************************/
char text_strings[10][20] = {
         {"first string"},
         {"second string"},
         {"third string"},
         {"fourth string"},
         {"fifth string"},
         {"sixth string"},
         {"seventh string"},
         {"eighth string"},
         {"ninth string"},
         {"tenth string"}
};

/*******************************************************/
/* Labels for different Prop Gadgets                   */
/*                                                     */
/*******************************************************/

   struct IntuiText labels[4] = {
         {2,0,JAM2,5,15,NULL,
                  (UBYTE *)"scroll strings",NULL},
         {2,0,JAM2,220,15,NULL,
                  (UBYTE *)"Prop fuzzy",&labels[0]},
         {2,0,JAM2,470,15,NULL,
                  (UBYTE *)"image/highlight",&labels[1]},
         {2,0,JAM2,200,108,NULL,
                  (UBYTE *)"free Vert and Horiz",&labels[2]}
};


/*********************************/
/* Declare NewWindow structure. */
/*********************************/

struct NewWindow NewControlWindow = {
   20, 20,              /* start LeftEdge, TopEdge */
   600, 160,            /* start Width, Height */
   2, 3,                /* DetailPen, BlockPen */
                        /* IDCMP FLAGS */
   GADGETUP | GADGETDOWN | CLOSEWINDOW |
                  MOUSEMOVE | MENUPICK,
                        /* Flags */
   WINDOWDRAG | WINDOWDEPTH | WINDOWCLOSE | ACTIVATE,
   &PropGadget_free,    /* Pointer to FirstGadget */
   NULL,                /* no pointer to first CheckMark */
   (UBYTE *)"Proportional Gadgets",/* Title */
   NULL,                /* no Pointer to Screen */
   NULL,                /* no Pointer to BitMap */
   20, 20,              /* Min size (no size allowed) */
   600, 160,            /* Max size (no size allowed) */
   WBENCHSCREEN         /* Type of screen */
};

/*******************************************************/
/* Main program                                        */
/*******************************************************/

main()
{
   struct MenuItem *ItemAddress();
   ULONG Signals, MIClass, MICode, itemnum;
   APTR MIAddress;
   LONG gad_pos, real_pos;
   SHORT MouseX, MouseY, first_line;
   BOOL fix_strings;

/* Open libraries */

   if (!(IntuitionBase = (struct IntuitionBase *)
    OpenLibrary("intuition.library",
                  (LONG)LIBRARY_VERSION)))
   {
      printf("Can't open the intuition library\n");
      MyCleanup();
      exit(FALSE);
   }

   if (!(GfxBase = (struct GfxBase *)
    OpenLibrary("graphics.library",
                  (LONG)LIBRARY_VERSION)))
   {
      printf("Can't open the graphics library\n");
      MyCleanup();
      exit(FALSE);
   }

/*******************************************************/
/* Set Prop Gadget variables for knob and motion       */
/* of knob.                                            */
/*******************************************************/

   g_prop_scroll.Flags = AUTOKNOB | FREEVERT;
   g_prop_scroll.VertBody = (ULONG)(visible_lines * 0xFFFF)
                                    / total_lines;

   g_prop_fuzzy.Flags = AUTOKNOB | FREEVERT;
   g_prop_fuzzy.VertBody = 0x3000;

   g_prop_image.Flags = FREEVERT;
   g_prop_image.VertBody = 0x3000;

   g_prop_free.Flags = FREEVERT | FREEHORIZ | PROPBORDERLESS;
   g_prop_free.VertBody = 0x7FFF;
   g_prop_free.HorizBody = 0x7FFF;

/*******************************************************/
/* Call routine to assign strings to the IntuiText     */
/* structures. We'll start at the top of the list.     */
/*******************************************************/

   SetString(0);
   PadString();

/*******************************************************/
/* Open window in which to display Proportional Gadgets. */
/*******************************************************/

      if (!(ControlWindow =
         (struct Window *)OpenWindow(&NewControlWindow)))
      {
         printf("Couldn't open the control window.\n");
         MyCleanup();
         exit(FALSE);
      }

   PrintIText(ControlWindow->RPort, &labels[3], 0L, 0L);

/*********************************************/
/* Write out the strings to the window.      */
/*********************************************/
   PrintIText(ControlWindow->RPort, &IntuiStr[4], 0L, 0L);

/*******************************************************/
/* Loop forever until user clicks Close Gadget on window.*/
/*******************************************************/
   for (;;) {  /* wait for a signal and process it */

   fix_strings = FALSE;

      Signals = Wait(1L <<
```

continued....

Avatex 1200

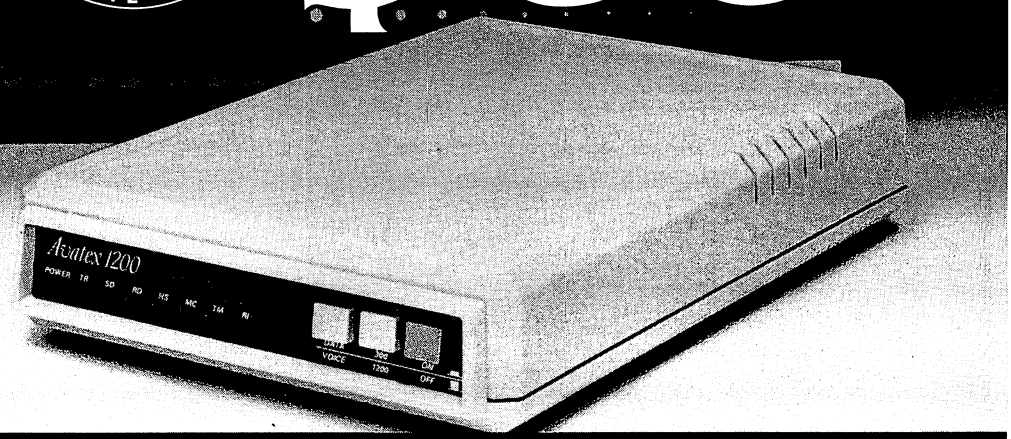
MODEM

THIS MODEM WORKS WITH ANY AMIGA.

- Direct Connect
- External AC Adapter
- 8 LED Status Indicators
- Auto Answer and Dial
- Hayes Compatible AT Command Set
- Bell 103/212A Compatible
- Tone or Pulse Dialing
- Telephone Connection Jack
- Standard DB25RS 232-C Connector
- FCC Approved

COMPUTER SHOPPER Best Buy OF THE YEAR

$85



FREE! Communication Software & Compuserve Access Time with each MODEM.

We give you a great price . . .
+ TOLL FREE ASSISTANCE
+ IMMEDIATE DELIVERY
+ NO CREDIT CARD SURCHARGE
+ NO SERVICE OR HANDLING FEES
+

We lowered our price.

Thousands of people are now enjoying this high quality Modem. It is full of features and works with any computer system. We are offering this special price for a limited time only.

FREE SHIPPING!

MEGATRONICS

To Order
CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

800-232-6342
INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.

MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

```
        ControlWindow->UserPort->mp_SigBit);

        /* Process the Intuition message. */
        while (MyIntuiMessage=(struct IntuiMessage *)
                    GetMsg(ControlWindow->UserPort))
        {
        /* Get all the needed info and reply to message */
        MIClass = MyIntuiMessage->Class;
        MICode = MyIntuiMessage->Code;
        MIAddress = MyIntuiMessage->IAddress;
         MouseX = MyIntuiMessage->MouseX;
        MouseY = MyIntuiMessage->MouseY;

        ReplyMsg(MyIntuiMessage);

        /* Determine what the message was. */

          switch (MIClass)
           {
             case GADGETUP:
                 first_line = ((ULONG)(total_lines -
                   visible_lines) * g_prop_scroll.VertPot
                   + (1L<<15)) >>16;

                 SetString(first_line);
                 PadString();
                 PrintIText(ControlWindow->RPort,
                          &IntuiStr[4], 0L, 0L);
             break;
             case GADGETDOWN:
             break;
             case MOUSEMOVE:

/*****************************************************/
/* We'll set a flag to tell ourselves to update the */
/* strings. This way we don't update them for EVERY */
/* mousemove message.                                */
/*****************************************************/
                 fix_strings = TRUE;
             break;

/* User clicked close Gadget. */
             case CLOSEWINDOW:

/* Reply to any outstanding messages. */
             while (MyIntuiMessage =(struct IntuiMessage *)
                   GetMsg(ControlWindow->UserPort))
                    ReplyMsg(MyIntuiMessage);
             MyCleanup();
             exit(TRUE);
             break;

           } /* switch */
         } /* while */

/*****************************************************/
/* Check if we need to update list of strings.       */
/* Call routine to assign strings to the IntuiText   */
/* structures.                                       */
/*****************************************************/
         if (fix_strings)
         {
         first_line = ((ULONG)(total_lines - visible_lines) *
                  g_prop_scroll.VertPot + (1L<<15)) >>16;
         SetString(first_line);
         PadString();
         PrintIText(ControlWindow->RPort,
                  &IntuiStr[4], 0L, 0L);
         }

    } /* for */
} /* main */

MyCleanup()
{
   if (ControlWindow) CloseWindow(ControlWindow);
   if (GfxBase) CloseLibrary(GfxBase);
   if (IntuitionBase) CloseLibrary(IntuitionBase);
}

/*****************************************************/
/* This routine assigns a string to each of the      */
/* IntuiText structures we will display.             */
/*****************************************************/
SetString(index)
WORD index;

{
int i;

   for(i=0; i<5; i++)
         IntuiStr[i].IText = (UBYTE *)text_strings[index +
i];
}

/*****************************************************/
/* This routine will pad each string we have decided */
/* to display with blanks. In this example it is     */
/* redundant - we will blank out the same strings over  */
/* over. In a real application, however, we would, most  */
/* likely, be changing the text in the text_strings array*/
/* and would therefore need to re-blank them.
*/
/*****************************************************/
PadString()
{
int len,i,j;
   for (i=0; i<5; i++)
   {
      len = strlen(IntuiStr[i].IText);
      for (j = len; j < MAXLEN - 1; j++)
      {
         * (IntuiStr[i].IText + j) = ' ';
      }
      * (IntuiStr[i].IText + MAXLEN - 1) = '\0';
   }
}
```

•AC•

68000 Assembly Language Programming *on the Amiga™*

by Chris Martin

Last month, we discussed condition flags and conditional statements. This month, I will list all the assembly language instructions and I will give examples of each. All you have learned from previous articles will be of use to you here, in understanding the various forms of each instruction. I will group each statement into one of the 7 basic categories: Data Movement, Arithmetic, BCD Arithmetic, Bit Manipulation, Logical, Program Control and System Control.

Mode	Symbol	Data	Control	Memory	Alterable	Example
Data Reg. Direct	Dn	x			x	D1
Addr. Reg. Direct	An				x	A4
Absolute	nnnnn	x	x	x	x	90324
Immediate	#imm	x	x			#100
Addr. Reg. Indirect	(An)	x	x	x	x	(A2)
" with predecrement	-(An)	x		x	x	-(A6)
" with postdecrement	(An)+	x		x	x	(A1)+
" with displacement	d(An)	x	x	x	x	24 (A3)
" with index	d(An,Ri)	x	x	x	x	3 (A2,D4)

Throught the list, I will classify the addressing modes under the following categories: Data, Control, Memory and Alterable. See last month's *Amazing Computing.*

The following codes, based on the above categories, will be used in the list of instructions:

[a]	Any addressing mode.
[aa]	Alterable addressing mode.
[ca]	Control addressing mode.
[da]	Data mode.
[caa]	Control or alterable mode.
[daa]	Data or alterable mode.
[maa]	Memory or alterable mode.

Data Movement

EXG Rn, Rn — Exchanges one 32-bit reg with another.
LEA address, An — Loads an address into address reg "An".
LINK An, #imm — #imm specifies stack size and links the current stack with another stack starting at address in An.
UNLK An — Reverses the LINK instruction.
MOVE [a], [daa] — Move data (.B, .W, .L)
MOVEA [a], An — Move address (.W, .L)

MOVEM [reg. list],-(An)
MOVEM [reg. list],[caa]
MOVEM (An)+, [reg. list]
MOVEM [ca], [reg. list]
— Move multiple registers into memory locations or vice-versa. (.W, .L)

MOVEP Dn, d(An)
MOVEP d(An), Dn
— MOVEP copies 2 or 4 bytes of data from a data register into "alternate" data locations, or vice-versa. (.W, .L)

MOVEQ #imm, Dn — Quick move immediate value to data register. (.L)
PEA [ca] — Calculates the address of the operand and pushes it onto the stack. (.L)
SWAP Dn — Exchanges the values of the high and low words of the specified data register. (.W)

These instructions are basically used to move data from one place to another. Don't worry if you feel lost. Stick with it. You'll understand much more when we actually start programming.

continued...

Arithmetic

These instructions are used strictly with binary integers: With the instructions of the form XXX a, b. The operation is performed between a and b, then the result is stored in b.

ADD [a],Dn ADD Dn, [maa]	binary addition: add binary	.B, .W, .L
ADDA [a],An	add address	.W, .L
ADDI #imm,[daa]	add immediate	.B, .W, .L
ADDQ #imm,[aa]	add quick	.B, .W, .L
ADDX Dn,Dn ADDX -(An),-(An)	add extended	.B, .W, .L
CLR [daa],Dn	clear, set to 0.	.B, .W, .L
CMP [a],Dn	compare [a] with Dn	.B, .W, .L
CMPA [a],An	compare address	.W, .L
CMPI #imm, [daa]	compare immediate	.B, .W, .L
CMPM (An)+,(An)+	compare memory	.B, .W, .L
DIVS [da],Dn	binary division: signed	.W
DIVU [da],Dn	binary division:unsigned	.W
EXT Dn	extend the sign bit (bit 7) into bit 8 to bit 15, or bits 16 to 31, depending on the size of the operation	.W, .L
MULS [daa],Dn	binary multiply: signed	.W
MULU [daa],Dn	binary multiply: unsigned	.W
NEG [daa]	make negative.	.B, .W, .L
NEGX [daa]	negate and extend sign	.B, .W, .L
SUB [a],Dn SUB Dn,[maa]	binary subtract	.B, .W, .L
SUBA [a],An	subtract from address	.W, .L
SUBI #imm,[daa]	subtract immediate	.B, .W, .L
SUBQ #imm,[aa]	subtract quick (#imm must be 1 to 8)	.B, .W, .L
SUBX -(An),-(An) SUBX Dn,Dn	subtract with sign extend. The "extended sign" bit is added to the first operand.	.B, .W, .L
TAS [daa]	TAS tests byte contained in the specified memory location & sets negative flag or the zero flag accordingly.	.B
TST [daa]	compares the operand with 0 and sets the flags accordingly.	.B, .W, .L

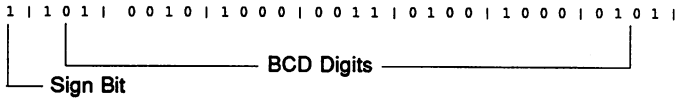
BCD Arithmetic

There are some instances where binary representation of numbers is inconvenient in of storing and transmitting data. The solution is to use the Binary Coded Decimal (or BCD) arithmetic system. In the BCD system, digits are separately represented and stored as a set of nibbles, (half-bytes).

BINARY	BCD	BINARY	BCD
0000	0	1000	8
0001	1	1001	9
0010	2	1010	-
0011	3	1011	-
0100	4	1100	-
0101	5	1101	-
0110	6	1110	-
0111	7	1111	-

Notice that there are 16 different ways to present 4 bits, but only 10 digits needed in BCD arithmetic. This will cause some problems when adding or subtracting BCD numbers, which I will explain soon.

The number -2834.85, for example, might be stored like:



One problem we encounter in BCD includes certain bit combinations which do not create a nibble representing a digit. For example, consider the addition of 4 and 8 to make 12:

```
    0000 0100  =  4
+   0000 1000  =  8
    0000 1100  =  12  in binary,  BUT NOT IN BCD!
```

In BCD, the number 12 would be represented by:

```
0001 0010

1    2
```

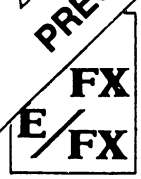
Because of these types of problems, a set of instructions which automatically corrects this is reserved in the 68000. Here are the BCD instructions.

BCD Instructions

ABCD Dn, Dn ABCD -(An),-(An)	BCD addition	.B
SBCD Dn,Dn SBCD -(An),-(An)	BCD subtraction	.B
NBCD [daa]	Negate decimal	.B

ACS PRESENTS

PRODUCTIVITY SOFTWARE



STATION MANAGER E/FX

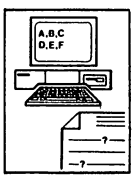
- Sophisticated television station graphics and animation system.
- Easy to use "storyboard" based graphics engine makes the creation of still or animated graphics sequences quick and simple.
- Uses high resolution (672x444), double buffered animation for super smooth movements.
- Extremely flexible advanced color cycling functions.
- Over 40 different wipes, dozens of backgrounds, hundreds of brushes.
- Fully mouse driven.
- Works with any high resolution AMIGA graphics packages.
- *ONLY* $295.00 with remote control *ONLY* $395.00

STATION MANAGER WEATHER GRAPHICS

- Full television station weather graphics system.
- Works in conjunction with E/FX to provide stunning graphics and animation. (see E/FX)
- Dozens of high resolution (672x444) IFF weather maps and backgrounds.
- All maps and backgrounds are completely user-customizable with the addition of any hi-res AMIGA drawing package.
- Down-load software allows reception of maps and weather data from all the most popular weather services.
- Map customization services available.
- *WEATHER GRAPHICS* $295.00
- *WEATHER GRAPHICS DELUXE* $995.00 (Includes E/FX, remote control, Deluxe Paint* and customized maps)

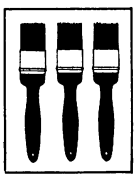
QUIZ MASTER

- Allows teachers to create their own lessons in any subject, or edit and personalize MUSIC STUDENT I and II.
- Available for AMIGA and MS-DOS. (MS-DOS requires 2 drives and graphics board.)
- Permits 8 levels (240 questions). Up to 40 questions per lesson. Up to 9600 questions all on one disk.
- Any question format: T/F, multiple choice, fill in the blank, and answer the question.
- Supports sound, music and graphics.
- *ONLY* $79.95



QUICK MERGE

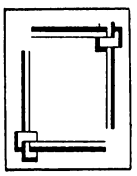
- The first AMIGA mail merge program that supports ALL Textcraft type styles, margin settings and justifications.
- Works with data exported from any AMIGA database or created in any text editor.
- User definable field and record separators.
- Raw, Final or Draft output modes.
- Print to Disk or Printer with complete control over page break separators.
- Variables may be used an unlimited number of times, ANYWHERE with your Textcraft document.
- *ONLY* $59.95!



BRUSH-WORKS Volume I

BUSINESS PRESENTATION GRAPHICS CLIP-ART

- Over 100 unique symbols and images.
- No conversion necessary—all brushes have been drawn in high, medium and low resolution modes.
- AMIGA IFF compatible—use with any AMIGA graphics package.
- *ONLY* $29.95!



BRUSH-WORKS Volume II

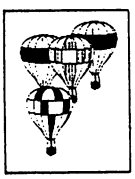
BOXES AND BORDERS

- Over 500 different borders and text box combinations.
- Designed especially for use in desktop publishing packages such as Page Setter*.
- Themes include: historical (from Roman to Modern), seasonal, architectural, theatre, heraldic, caption, landscape and industrial design.
- *ONLY* $29.95!

BRUSH-WORKS Volume III

FONTS AND FIXTURES

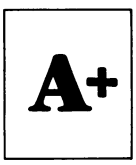
- 25 all new AMIGA fonts.
- Each font works in all AMIGA resolution modes.
- Powerful 'FONT MAKER' allows you to turn digitized or scanned IFF pictures into font sets in just minutes!
- Designed for use in desktop publishing and business presentation graphics.
- *ONLY* $29.95!



BRUSH-WORKS Volume IV

DESKTOP PUBLISHING CLIP-ART

- Over 200 pieces of clip-art designed especially for AMIGA desktop publishing.
- Full IFF compatibility allows all images to be used not only in desktop publishing but in any medium resolution AMIGA graphics package.
- Themes include: holidays, seasons, captions, people, sports, cartoons and MUCH more!
- *ONLY* $29.95!



GRADE MANAGER

- Comprehensive electronic gradebook.
- Up to 100 students with 100 grades each.
- Handles extra credit and incompletes.
- Automatic conversion to 100% point scale.
- Prints rosters, gradebook sheets, grade lists, progress reports and more.
- 8 weighting categories.
- *ONLY* $89.95 (Amiga) $69.95 (Commodore).

MUSIC STUDENT I

- Music theory concepts.
- 178 lessons on one disk.
- Intervals, triads, scales, terms, symbols, and more.
- 8 levels—beginning through intermediate.
- *ONLY* $59.95.

ORDER TODAY!
MasterCard and Visa accepted.

ORDERS ONLY: 1-800-538-1032 EXT. 7373
IN MISSOURI CALL: 1-800-492-5067 EXT. 7373

Associated Computer Services
1306 E. Sunshine, Springfield, Missouri 65804
Inquiries: (417) 887-7373

*These are trademarks of Commodore-Amiga, Microsoft, Commodore Intl., Gold Disk and Electronic Arts.

FOR AMIGA COMPUTERS

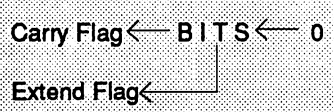
Bit Manipulation

These instructions are used to test, clear and change specific bits. They all operate in similar manners. The bit number is specified in the source operand, Dn for one of 32 bits or #[imm] for one of 8 bits. Following this, the bit is operated on accordingly and the ZERO flag is set, depending on the nature of the tested bit (0 or 1).

BCHG Dn,[daa] BCHG #imm,[daa]	Bit test and change. Bit of 1 changed to 0 (and vice-versa)	.B .L
BTST Dn,[da] BTST #imm,[da]	Bit test. Bit remains unchanged	.B .L
BSET Dn,[daa] BSET #imm,[daa]	Bit first tested then always set to 1.	.B .L
BCLR Dn,[daa] BCLR #imm,[daa]	Bit tested then always set to 0.	.B .L

The following instructions are used to manipulate sets of bits in various ways. The source operand (the first) is taken, manipulated and stored in the destination operand.

ASL Dn,Dn Arithmetic shift left works as follows:



The leftmost bit is sent to the carry and extend flags, while a 0 replaces the rightmost bit.

ASL #imm,Dn .B .W
ASL [maa]

ASR Dn,Dn ASR #imm,Dn ASR [maa]	Arithmetic shift right. Works in opposite manner as above. except leftmost bit is replaced by a duplicate of itself.	.B .W .L
LSL Dn,Dn LSL #imm,Dn LSL [maa]	Logical shift left.	.B .W .L
LSR Dn,Dn LSR #imm,Dn LSR [maa]	Logical shift right. Unlike ASR, a 0 is moved into the leftmost bit position.	.B .W .L
ROL Dn,Dn	Rotate bits left works in the following manner:	

The carry bit and the rightmost bit are replaced with the leftmost bit.

ROL #imm,Dn .B .W .L
ROL [maa]

ROR Dn,Dn ROR #imm,Dn ROR [maa]	Rotate right, opposite of above.	.B .W .L
ROXL Dn,Dn ROXL #imm,Dn ROXL [maa]	Same as ROL, except EXTEND BIT is also set with leftmost bit.	.B .W .L
ROXR Dn,Dn ROXR #imm,Dn ROXR [maa]	Same as ROR, except EXTEND BIT is also set with rightmost bit.	.B .W .L

Logical Instructions

There are three logical operators: AND, OR and EOR. AND takes a source and a destination operand and returns a result in which each bit is set, if both the corresponding bits in both the source and the destination are set. OR sets bits if either the source or destination bits are set. Finally, EOR sets bits if either source or destination bits are set, BUT NOT BOTH.

AND	OR	EOR
10101010	10101010	10101010
01101101	01101101	01101101
00101000	11101111	11000111

AND [da],Dn AND Dn,[maa]	AND bits.	.B .W .L
ANDI #imm,[daa]	AND immediate	.B .W .L
OR [da],Dn OR Dn,[maa]	OR bits.	.B .W .L
ORI #imm,[daa]	OR immediate	.B .W .L
EOR Dn,[daa]	EOR bits.	.B .W .L
EORI #imm,[daa]	EOR immediate	.B .W .L
NOT [daa]	Reverts all bits. 1 becomes 0, 0 becomes 1.	.B .W .L

Program Control

There are three types of statements that have many variations based on conditions. They are Scc (Set from condition), Bcc (Branch on condition), and DBcc (Decrement and Branch on condition). For example, BEQ means Branch if equal. Here is a list of the conditional suffixes:

CC	carry clear	if C=0
CS	carry set	if C=1
EQ	equal	if Z=1
GE	greater or equal	if either (N=1 and V=1) or (N=0 and V=0)
GT	greater	if either (N=1 & V=1 & Z=0) or (N=0 & V=0 & Z=0)
HI	high	if C=0 and Z=0
LE	less or equal	if (N=1 & V=0) or (N=0 & V=1) or Z=1
LS	low or same	if C=1 or Z=1
LT	less than	if either (N=1 & V=0) or (N=0 & V=1)
MI	minus	if N=1
NE	not equal	if Z=0
PL	plus	if N=0
VS	overflow	if V=1
VC	not overflow	if V=0

Scc [daa]	Set from condition. B Sets a destination byte for a specific condition: 255 is set if true, 0 if false.
Bcc <label>	Branches to Label on a certain condition.
BRA <label>	Branch always. Completely redirects execution.
BSR <label>	Same as BRA, except execution is redirected to a subroutine. The address of this subroutine is sent to the stack (which the programmer may examine at any time).
DBRA <label>	Decrement and branch, like BRA.
JMP [ca]	Jumps to address and redirects execution of the program.
JSR [ca]	Jumps to a subroutine. When RTS is encountered, the program jumps back to the location where the JSR was issued.
RTE	Return from exception subroutine
RTR	Return from a sub and restore the CCR.
RTS	Return from subroutine.

System Control

These instructions perform miscellaneous jobs (which we usually don't need to worry about).

RESET	Resets all external devices.
STOP #imm	Loads an immediate value into the status register and stops execution of the program, until an interrupt of sufficient priority occurs.
CHK [da],[da]	Check against bounds. The source operand is the boundary value and the destination is the value to be checked.

Traps are system routines that may be called at any time. Parameters may be passed to the routines through the data registers.

TRAP #imm	TRAP forces execution of a trap routine numbered in the range or 0 to 15.
TRAPV	Executes the TRAPV exception routine.

Whew! That was a long article! Now that you have a reference guide to all of the 68000 instructions, we can start on some programs. Don't worry if this seems like a lot for a beginner - once you see actual examples of programming in assembly, it will be much simpler. Until next month...

•AC•

Amiga Developers

This space could have been yours at a rate you wouldn't believe. Amazing Computing™ is the Amiga users' magazine, dedicated to helping the Amiga user get the most out of his machine. Amazing Computing™ is now sold at over 1200 locations worldwide and is read by users who want to do more with their Amigas. These Amiga users are searching for an application program or hardware item you have developed.

If you are an Amiga Developer who is trying to get the most cost effective advertising for a new product (without running a budget like the U.S. Government) then give us a call. If money is no object, we're certain other magazines will also be glad to help!

Contact:

John Fastino
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722
(617) 678-4200

Amazing Computing™: Your Resource to the Commodore Amiga™

The Amicus Network™

Commodore shake-ups, Microfiche Filer, ASDG RRD, New AMICUS and Fish disks

By John Foust

My phone was ringing almost continuously for a few days in April. It started in drips and splashes. The word was out: some kind of shakeup was afoot within Commodore. Then, a gush of juicy gossip. President and CEO Thomas Rattigan was out, along with several other top managers. The inside skinny on the whole affair never made it to public light, I suspect, judging by the tidbits that floated by. A few days later, the news went public. The stock price wavered and then all was calm.

It is all water under the bridge. In short, Commodore has a new chief executive officer, Irving Gould. He also remains as chairman of the board. Alfred Duncan replaced Nigel Shepard in the U.S. operations general manager spot. Richard McIntyre now fills a new general sales manager position. Duncan and McIntyre have good reputations within Commodore. They both held high-ranking positions within Commodore Canada.

New pricing

In the last six weeks, the price of the Amiga 2000 has jumped from $1495 to $1695 and finally, to an official announcement of $1995 retail. Some Amiga owners gloomily watched their upgrade hopes rise out of sight. On the bright side, the new pricing could take advantage of a well-known phenomena in the MS-DOS market. A product at a higher price is perceived as a greater value. Lotus 1-2-3 retails for several hundred dollars, yet exact work-alikes sell for less than $50. If they can sell any at all, Commodore will make more money at the higher price.

This pricing structure also opens a gap in the product line. It makes more sense that a cost-reduced Amiga 1000 will be made, an Amiga 1000 box with AmigaDOS 1.2 in ROM and a separate keyboard. It would fit in the intermediate price spot between the Amiga 500 and the Amiga 2000. A friend noted that the price of a Commodore 64, with monitor and disk drive, is almost the same price as a similarly configured Amiga 500 system. This spells the future of the Commodore 64, I am sure.

At the Winter Consumer Electronics Show, when asked "When will you advertise?," Commodore proudly announced they had switched advertising agencies. This wasn't an answer to the question, of course, but it served to deflect a serious question. Cynics countered that the act of change would be disruptive, in itself, and introduce even longer delays in the appearance of Amiga advertisements.

I haven't seen an Amiga advertisement in months. Like many Amiga owners, I wish Commodore would advertise more. I sometimes get the feeling that Amigas are sold only because of word-of-mouth recommendations from present owners, arguably a powerful force in any market, but certainly no path to record sales. Hopefully, Commodore will advertise the Amiga 500 as much as they did the Commodore 64. Many people agree that the Commodore 64 succeeded because of heavy advertising and not because of any technical superiority over the competing Atari computers of the same period.

On the Well network, there was comedic discussion of Amiga advertising. Someone posted alternative scripts for Amiga commercials, mostly featuring the crew of the Enterprise from Star Trek. After all, if IBM can make a connection between new, incompatible computers and the cast of M.A.S.H, why not let Kirk and Spock plug the Amiga? William Shatner did feature in advertisements for an earlier series of Commodore computers.

Commodore is listening to Amiga owners, at least in a small way. While writing this, I got a phone call. The caller identified themselves as "Commodore." First, they wanted to confirm that I bought an Amiga recently. I answered, yes, but not recently; I got my machine in 1985. They wanted to know if I was happy with it. I answered in the affirmative. I was so surprised by this call that I asked the person holding on my other line to hang up. I wanted to find out more. I turned the tables on them and started asking them questions.

The call was from Debbie Brault, an account rep at Commodore West Chester. Under a project called SPIFF, she was conducting a simple market survey by calling people on a list of Amiga software buyers. Last, she wanted to know if I would buy another Amiga. Again, I answered yes, I should be buying both an Amiga 500 and an Amiga 2000 this year. It turns out that a significant number of people answer this question positively. There are people with more than one Amiga.

Microfiche Filer

While on vacation on the East Coast, I attended a meeting of the general Amiga group of the Boston Computer Society. The BCS Amiga group is quite active, one of the largest in the country. They have an active technical group of developers, as well as a general group. Many BCS members get together informally at other times, as well as attending both meetings.

continued...

The BCS tech group includes Charlie Heath, co-moderator of the BIX Amiga conference and developer of TxEd and Fastfonts, Bob Page and Rich Miner of Zoxso, Gary Samad of Software Visions, Jeff Arnold of the Golden Hawk MIDI interface and Harriet and Bob Tolly of Tollysoft. Harriet writes the Intuition Gadget tutorials in this magazine. There are others in this tech group whose products are not yet announced, but I think the day will come when their names are commonly recognized.

The featured speaker at the meeting was Gary Samad of Software Visions, presenting his new product, Microfiche Filer. I'd describe it as a visually-oriented database. It is a product that takes advantage of the Amiga, and this market needs more products.

The central paradigm of the program is the microfiche card, a popular database format in the analog world. The contents of the current view into the database is always visible as a condensed version in one corner. Each record of the database is reduced to a few pixels and records are tiled flat against each other, like the records on a microfiche card. By positioning the mouse cursor over the small version, a magnified view of that record is brought up on the larger portion of the screen. It has a very flexible interface for everything. All the forms for entering, viewing and sorting the database are Intuition-driven and very free-form for positioning text and graphics.

Of course, it is impossible to read each tiny record on the reduced version of the database, but each record can be read with the magnifying glass. If you select records from the database - for example, all the people who live in a certain state - these records are highlighted in the tiny overview window.

The neatest feature of this reduction and magnified view is that IFF pictures or brushes are valid data for fields in the database and the picture is shown in reduced form within each displayed record. This looks like a great way to organize an IFF collection or set up an employee database that includes small digitized pictures... I don't think Samad knows the full extent to which this program will be used and I'm sure he'll be swamped with requests for a more complex version.

The audience response to Microfiche Filer surprised Samad. "I expected a lot of people to say "That's neat," but I didn't expect a standing ovation." Microfiche Filer is a very interesting product. A review will appear in an upcoming issue of *Amazing Computing*.

RRD

The ASDG recoverable RAM disk is now a standard part of the startup-sequence on many Amiga Workbench disks. ASDG Inc. is primarily an Amiga hardware manufacturer, but now they have moved into software that enhances hardware as well. The recoverable RAM disk (or RRD) was written by ASDG president Perry Kivolowitz. The files in the RRD are preserved across a warm boot. In other words, the files in the RAM: disk are intact after most visits from the Guru.

Files in the standard AmigaDOS RAM disk disappear in a crash, even though the data is preserved in memory. Most crashes do not destroy the contents of memory, so the data can be rescued. The memory chips themselves retain all their data while the power is on. On startup, the RRD checks to see if memory has been altered irrepairably. If not, it recovers the files in the RAM disk.

This product has been a life-saver for many Amiga users. Programmers love it. Most developers have large RAM expansions and they load the C 'include' files to the RAM disk to speed the program development process. Before the RRD, each crash meant a patient wait while the startup-sequence copied the 'include' files back to the RAM: disk because the files were destroyed. The RRD is available on Fish disk 58.

When I first heard of the recoverable RAM disk, in the fall of 1986, it was only available bundled with the RAM expansion hardware made by ASDG. If you bought a memory board, you got the recoverable RAM disk. I encouraged Kivolowitz to sell it separately because I knew it would be enormously popular as a commercial product. Instead, Kivolowitz made it shareware in January 1987. This means it can be freely distributed, for example, on a public domain disk. He did limit the shareware version to 2 megabytes of storage. If you buy an ASDG memory card, you get a version that can use all the memory you have.

In typical shareware fashion, the documentation asks for a payment if you use the software. The RRD documentation asks for $10. As I predicted, RRD is very popular. As of this writing, more than 250 people have voluntarily paid for the RRD. "That number is very exceptional for shareware," according to Kivolowitz. I agree. In fact, 250 would be a respectable number of sales for a commercial Amiga product of technical nature. (This is a sad and little-known fact. Some well-known Amiga products have only sold hundreds of copies.)

Kivolowitz is proud of the RRD and many respondents were pleased with it, as well. "Surprisingly, a lot of people sent in more than $10." He attributes this to two factors. "People appreciate the product and respect the work that went into it." He also mentioned that registered RRD owners will get a surprise in the mail this summer. I'd recommend register your copy of RRD as soon as possible.

Why did he develop the RAM disk? He heard a rumor that another hardware developer was developing a recoverable RAM disk and he thought ASDG would need one to compete in the market. It turned out the rumor was false. Kivolowitz thanks this company for giving him the impetus to develop the product.

A new version of the RRD should be available at this time. It is twice as fast. Being re-written in assembler, it is half the size. This version also keeps track of the date, so the system time will be reset in a linear fashion on reboot, instead of using the last file time from an actual disk in the system.

ASDG has another software product out now. It is a floppy speedup program called Facc, short for "floppy accelerator." It is priced at $34.95. It speeds access to frequently floppy files by storing recently accessed data in memory. Traditionally, only a certain amount of data is kept in memory; this is what ADDBUFFERS does to increase floppy speed.

It is difficult to benchmark cacheing schemes like Facc and ADDBUFFERS. At best, access to a floppy becomes almost as fast as RAM disk. Facc has some advantages over ADDBUFFERs or a RAM disk. According to Kivolowitz, it is much faster than ADDBUFFERS at finding data. It uses memory space more efficiently. A RAM disk holds everything, while the Facc only holds the stuff you use recently. ADDBUFFERS uses precious CHIP memory, while Facc uses more expendable FAST memory. Facc has an input window to view its efficiency and the number of buffers can be adjusted for peak use.

Timesaver

A strange tube arrived in the mail this week. Inside was an even stranger product. According to the manual, the Timesaver from C Ltd. is a "macro-clock... for lack of a better description." On the surface, it is a very non-obtrusive battery-backed clock. It fits in a cubbyhole on the underside of your Amiga. It acts as a pass-through for the keyboard. It has a small microcomputer controller inside. All keystrokes are sent to this computer, re-transmitted and possibly altered before being sent to the Amiga. It is completely unobtrusive and consumes no memory, but adds macro-key abilities to any program. I will review this product in a future issue.

Hard disks, con't.

In the wake of the *Amazing Computing* hard disk reviews, more information has come to light. I neglected to mention an extra feature in the PAL Jr. hard disk software. It has the ability to use FAST memory for its disk buffers, as opposed to other Amiga hard disks, which use CHIP memory for buffers. C Ltd. has improved its disk driver software, as noted in an addendum in the review. Supra has told me they have a new driver, but a copy did not arrive in time for this column.

Adding buffers to hard disks can improve performance in some cases. The AmigaDOS 'ADDBUFFERS' command allocates CHIP memory for buffers. If you create partitions on a hard disk, effectively splitting the hard disk into several hard disks of smaller size, each partition gets its own buffers and can eat up memory.

ADDBUFFERS uses CHIP memory because it uses a custom chip to move data between buffers. The custom chips can only access CHIP memory, as ADDBUFFERS consumes CHIP memory. The PAL Jr. unit has its own DMA chip and it can move data anywhere.

My pet peeve about power switch placement is apparently not unique; at least some disk manufacturers have heard it from

continued...

others as well. On the newer Supra drives, in particular the 30 megabyte model, the power switch is on the front of the drive unit. The next manufacturing run of PAL Jr. boxes will have the power switch towards the front, too.

The new fast file system has not arrived, at this writing. Rumors of other fast file systems have surfaced. One involves a hard disk project on the GEnie electronic network. A team of people has designed a low-cost do-it-yourself SCSI interface board and are now busy writing hard disk driver software. Another part of the project is creating an alternate file system for the Amiga. It would be much faster, according to its planners, in reads, writes and icon access.

Crashes

Amiga hard disks are still in a primitive stage, in some respects. With no standard hard disk controller, there might never be a backup program comparable to the MS-DOS world's Fastback, so Amiga hard disk backup will be slower than it could be. I am compulsive about backups. I absolutely despise losing critical files due to mistakes or mismanagement of a hard disk. With the advent of a standard controller for the Amiga 2000, the chances for fast, reliable backup get better. The current backup programs take quite a while to save the hard disk data to floppies and, of course, require a six-inch-stack of floppies to do so.

I also miss programs like the MS-DOS 'chkdsk' program. This program checks the integrity of the hard disk, insuring that the system data about sector allocation and used sectors all makes sense. The MS-DOS program isn't perfect, given the limited redundancy of the MS-DOS directory and file structure. With all the redundancy built-in to the Amiga file structure, an Amiga 'chkdsk' program would be much more reliable.

I've had several hard disk crashes on my Amiga and I can't decide which program might be writing bad data and causing the crashes. I've heard rumors about certain programs that crash hard disks, but I am afraid to name them without more definite proof. Formatting and restructuring the disk can take a long time. I've also lost files because I forgot to save a subdirectory of files to floppy before formatting.

New AMICUS disks

There are two new AMICUS disks. These disks are of special interest to public domain junkies. Using the public domain disk cataloging program called DiskCat, written by Ed Alford, I cataloged all the AMICUS disks from 1 to 20 and all the Fred Fish disks from 1 to 68 and put the catalogs on these two new disks.

DiskCat can sort the database in several ways. On AMICUS 21, the AMICUS catalog has been sorted by filename, as well as by disk number. By sliding a gadget on the display, you can look at all the files that begin with the letter "F," for example. AMICUS 21 has the Fish catalog sorted by name; AMICUS 22 has it sorted by disk. AMICUS 22 also has a short paragraph description of each program on the first 68 Fish disks.

I find these catalogs very useful, especially when I need to find a certain program. Of course, I have to remember the actual filename. DiskCat doesn't respond to searches such as "find all font programs," although looking at all the files that begin with 'F' gives a good chance of finding something. Printed listings take quite a while to search. The electronic listing, sorted by filename, also warns about updated files because all versions of the program will be close together in an alphabetized list.

If you have a lot of memory, you can concatenate these catalogs into a single database and re-sortthemt. This will save out to a large file, a single catalog that encompasses all the disks in a collection. This file is more than 400K for the Fish catalog. Re-sorting this catalog takes several hours, but the results are well worth it, if you have a lot of memory or a hard disk to store it.

I considered putting this larger catalog on an AMICUS disk, but I felt it wasn't of general interest. It could only be accessed by machines with extra memory, to include both sorted-by-name and sorted-by-disk catalogs would have encompassed two more AMICUS disks.

The documentation for DiskCat is also present on AMICUS 21 and 22. You can use it to catalog your own disks or the public domain library of your user group.

AMICUS 22 also has the printer driver generator by Joergen Thomsen. With it, you can create a working printer driver for nearly any printer. The program requires knowledge of the most intimate details of your printer, but it does work, if you supply the correct information. By modifying a text file with 349 parameters, the generator will make a printer driver with those characteristics. You can also use it to create variations on your present printer driver, by making it print special characters such as umlauted vowels and Spanish 'n's. Thomsen has become somewhat of an expert on printer drivers on the national networks.

New Fish disks

At press time, Fred Fish announced disks 69 to 74. If you've been following this column to hear about the new files in his collection, you must be aware of a new law of computing. This law, which I will call Foust's First Law of Public Domain Software, says that public domain software will be created faster than it can be described. This column can't keep up with the flow of Amiga public domain software, but I will try. With this in mind, I don't know if I'll ever have the space to describe each and every program on the newborn Fish disks. As space permits, *Amazing Computing* will continue to carry the condensed listing in the back of the magazine.

•AC•

The AMICUS & Fred Fish Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1

ABasic programs: Graphics

3DSolids	3d solids modeling program w/sample data files
Blocks	draws blocks
Cubes	draws cubes
Durer	draws pictures in the style of Durer
FScape	draws fractal landscapes
Hidden	3D drawing program, w/ hidden line removal
JPad	simple paint program
Optical	draw several optical illusions
PaintBox	simple paint program
Shuttle	draws the Shuttle in 3d wireframe
SpaceArt	graphics demo
Speaker	speech utility
Sphere	draws spheres
Spiral	draws color spirals
ThreeDee	3d function plots
Topography	artificial topography
Wheels	draws circle graphics
Xenos	draws fractal planet landscapes

ABasic programs: Tools

AddressBook	simple database program for addresses
CardFile	simple card file database program
Demo	multiwindow demo
KeyCodes	shows keycodes for a key you press
Menu	run many ABasic programs from a menu
MoreColors	way to get more colors on the screen at once, using aliasing
shapes	simple color shape designer Speakit speech and narrator demo

ABasic programs: Games

BrickOut	classic computer brick wall game
Othello	also known as 'go'
Saucer	simple shoot-em-up game
Speling	simple talking spelling game
ToyBox	selectable graphics demo

ABasic programs: Sounds

Entertainer	plays that tune
HAL9000	pretends it's a real computer
Police	simple police siren sound
SugarPlum	plays "The Dance of the Sugarplum Fairies"

C programs:

ATerm	simple terminal program, S-E
cc	aid to compiling with Lattice C
decvnt	opposite of CONVERT for cross developers
Dotty	source code to the 'dotty window demo
echox	unix-style filename expansion, partial S, O-D
fasterfp	explains use of fast-floating point math
FixDate	fixes future dates on all files on a disk, S-E
freedraw	simple Workbench drawing program,S-E
GfxMem	graphic memory usage indicator, S-E
Grep	searches for a given string in a file, with documentation
ham	shows off the hold-and-modify method of color generation
IBM2Amiga	fast parallel cable transfers between an IBM and an Amiga
Mandel	Mandelbrot set program, S-E
moire	patterned graphic demo, S-E
objfix	makes Lattice C object file symbols visible to Wack, S-E
quick	quick sort strings routine
raw	example sample window I/O
setface	turns on interlace mode, S-E
sparks	qix-type graphic demo, S-E

Other executable programs:

SpeechToy	speech demonstration
WhichFont	displays all available fonts

Texts:

68020	describes 68020 speedup board from CSA
Aliases	explains uses of the ASSIGN command
Bugs	known bug list in Lattice C 3.02
CLICard	reference card for AmigaDOS CLI
CLICommands	guide to using the CLI
Commands	shorter guide to AmigaDOS CLI commands
EdCommands	guide to the ED editor
Filenames	AmigaDOS filename wildcard conventions
HalfBright	explains rare graphics chips that can do more colors
ModemPins	description of the serial port pinout
RAMdisks	tips on setting up your RAM: disk
ROMWack	tips on using ROMWack
Sounds	explanation of the instrument demo sound file format
Speed	refutation of the Amiga's CPU and custom chip speed
WackCmds	tips on using Wack

AMICUS Disk 2

C programs:

alib	AmigaDOS object library manager , S-E
ar	text file archive program, S-E
fixobj	auto-chops executable files
shell	simple CLI shell, S-E
sq, usq	file compression programs, S-E
YachtC	a familiar game, S-E
Make	a simple 'make' programming utility, S-E
Emacs	an early version of the Amiga text editor, S-E-D

Assembler programs:

bsearch.asm	binary search code
qsort.asm	Unix compatible qsort() function,
source	and C test program
setjmp.asm	setjmp() code for Lattice 3.02
SVprintf	Unix system V compatible printf()
trees.o	Unix compatible tree() function, O-D

(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection. They are not here.)

John Draper Amiga Tutorials:

Animate	describes animation algorithms
Gadgets	tutorial on gadgets
Menus	learn about intuition menus

AMICUS Disk 3

C programs:

Xref	a C cross-reference gen., S-E
6bitcolor	extra-half-bright chip gfx demo, S-E
Chop	truncate (chop) files down to size, S-E
Cleanup	removes strange characters from text files
CR2LF	converts carriage returns to line feeds in Amiga files, S-E
Error	adds compile errors to a C file, S
Hello	window ex. from the RKM, S
Kermit	generic Kermit implementation, flakey, no terminal mode, S-E
Scales	sound demo plays scales, S-E
SkewB	Rubik cube demo in hi-res colors, S-E

AmigaBasicProgs(dir)

Automata	cellular automata simulation
CrazyEights	card game
Graph	function graphing programs
WitchingHour	a game

ABasIC programs:

Casino	games of poker, blackjack, dice, and craps
Gomoku	also known as 'othello'
Sabotage	sort of an adventure game

Executable programs:

Disassem	a 68000 disassembler, E-D
DpSide	shows a given set of IFF pictures, E-D
Arrange	a text formatting program, E-D

Assembler programs:

Argoterm	a terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and bwrite.c

addmem.c	add external memory to the system
bobtest.c	example of BOB use
consoleIO.c	console IO example
creaport.c	create and delete ports
creastdi.c	create standard I/O requests
creatask.c	creating task examples
diskio.c	example of track read and write
dotty.c	source to the 'dotty window' demo
dualplay.c	dual playfield example
flood.c	flood fill example
freemap.c	old version of 'freemap'
getbobs.c	tools for VSprites and BOBs
gfxmem.c	graphic memory usage indicator
hello.c	window example from RKM
inputdev.c	adding an input handler to the input stream
joystik.c	reading the joystick
keybd.c	direct keyboard reading
layertes.c	layers examples
mousport.c	test mouse port
ownlib.c, ownlib.asm	example of making your own library with Lattice
paratest.c	tests parallel port commands
seritest.c	tests serial port commands
serisamp.c	example of serial port use
prinintr.c	sample printer interface code
prtbase.h	printer device definitions
regintes.c	region test program
setlace.c	source to interlace on/off program
setparallel.c	set the attributes of the parallel port
SetSerial.c	set the attributes (parity, data bits) of the serial port
singplay.c	single playfield example
speechtoy.c	source to narrator and phonetics demo
timedely.c	simple timer demo
timer.c	exec support timer functions
timrstuf.c	more exec support timer functions
WhichFont.c	loads and displays all available system fonts

process.i and prtbase.i assembler include files:

autoreqstr.txt	warnings of deadlocks with autorequesters
consoleIO.txt	copy of the RKM console I/O chapter
diskfont.txt	warning of disk font loading bug
fullfunc.txt	list of #defines, macros, functions
inputdev.txt	preliminary copy of the input device chapter

License information on Workbench distribution license

printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1 v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

A demo of Intuition menus called 'menudemo', in C source

whereis.c	find a file searching all subdirectories
bobtest.c	BOB programming example
sweep.c	sound synthesis example

Assembler files:

mydev.asm	sample device driver
myfib.asm	sample library example
myfib.i	
mydev.i	
asmsupp.i	
macros.i	assembler include files

Texts:

amigatricks	tips on CLI commands
extdisk	external disk specification
gameport	game port spec
parallel	parallel port spec
serial	serial port spec
v1.1update	list of new features in version 1.1
v1.1h.txt	'diff' of include file changes from version 1.0 to 1.1

Files for building your own printer drivers, including dospecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon, and the 'saveilbm' program, to turn any screen into an IFF picture. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Marlian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newscaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a 'tan-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture seperately, and all together as separate, sizable screens.

AMICUS Disk 8

C programs:

Browse	view text files on a disk, using menus S-E-D
Crunch	removes comments and white space from C files, S-E
IconExec	EXECUTE a series of commands from Workbench S-E
PDScreen Dump	dumps Rastport of highest screen to printer
SetAlternate	sets a second image for an icon, when clicked once S-E
SetWindow	makes windows for a CLI program to run under Workbench S-E
SmallClock	a small digital clock that sits in a window menu bar
Scrimper	the screen printer in the fourth Amazing Computing, S-E

Amiga Basic Programs:

(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)

AddressBook	a simple address book database
Bell	draws a bell
Clead	program to convert Compuserve hex files to binary, S-D
Clue	the game, Intuition driven
ColorArt	art drawing program
DeluxeDraw	the drawing program in the 3rd issue of Amazing Computing, S-D
Eliza	conversational computer psychologist
Othello	the game, as known as 'go'
RatMaze	3D ratmaze game
ROR	boggling graphics demo
Shuttle	draws 3D pictures of the space shuttle
Spelling	simple spelling program
YoYo	wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse

Executable programs:

3Dcube	Modula-2 demo of a rotating cube
Aticon	sets a second icon image, displayed when the icon is clicked
AmigaSpel	a slow but simple spell checker, E-D
arc	the ARC file compression program, must-have for telecom, E-D
Bertrand	graphics demo
diskSalvage	prog. to rescue trashed disks, E-D
KwikCopy	a quick but nasty disk copy program: ignores errors, E-D
LibDir	lists hunks in an object file E-D
SaveILBM	saves any screen as IFF pic.E-D ??
ScreenDump	shareware screen dump prog, E only
StarTerm	version 2.0, term program, Xmodem E-D

Texts:

LatticeMain	tips on fixing _main.c in Lattice
GDiskDrive	make your own 5 1/4 drive
GuruMed	explains the Guru numbers
Lat3.03bugs	bug list of Lattice C version 3.03
MForgeRev	user's view of the MicroForge HD
PrintSpooler	EXECUTE-based print spool prog.

.BMAP files:

These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieeesingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9

Amiga Basic Programs:

FlightSim	simple flight simulator program
HuePalette	explains Hue, Saturation, and Intensity
Requester	ex. of doing requesters from Amiga Basic

ScrollDemo	demonstrates scrolling capabilities
Synthesizer	sound program
WorldMap	draws a map of the world

Executable programs:

Boing!	latest Boing! demo,with selectable speed, E
Brush2C	converts an IFF brush to C data instructions, initialization code, E
Brush2Icon	converts IFF brush to an icon, E
Dazzle	graphics demo, tracks to mouse, E
DeciGEL	assembler program for stopping 68010 errors, S-E-D
Klock	menu-bar clock and date display, E
life	the game of life, E
TimeSet	intuition-based way to set the time and date,
EMEmacs	another Emacs, more oriented to word processing, S-E-D
MyCLI	a CLI shell, works without the Workbench, S-E-D

Texts:

FnctnKeys	explains how to read function keys from Amiga Basic
HackerSln 'hacker'	explains how to win the game
ist68010	guide to installing a 68010 in your Amiga
PrinterTip	tips on sending escape sequences to your printer
StartupTip	tips on setting up your startup-sequence file
XtrnrReview	list of programs that work with the Transformer

Printer Drivers:

Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 **Instrument sound demos**

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11

C programs

dinutil	Intuition-based, CLI replacement
file	manager, S-E
cpri	shows and adjusts priority of CLI processes, S-E
ps	shows info on CLI processes, S-E
victax	displays Compuserve RLE pics, S-E

AmigaBasic programs

pointered	pointer and sprite editor program
optimize	optimization ex ample from AC article
calendar	large, animated calendar, diary and date book program
amortize	loan amortizations
brushtoBOB	converts small IFF brushes to AmigaBasic BOB OBJECTS
grids	draw and play waveforms
hilbert	draws Hilbert curves
madlib	mad lib story generator
mailtalk	talking mailing list program
meadows3D	3D graphics program, from Amazing Computing™ article
mousetrack	mouse tracking example in hires mode
slot	slot machine game
tictactoe	the game
switch	pachinko-like game
weird	makes strange sounds

Executable programs

cp	unix-like copy command, E
cls	screen clear, S-E
diff	unix-like stream editor uses 'diff' output to fix files
pm	chart recorder performances indicator

Assembler programs

cls	screen clear and CLI arguments example

Modula-2

trails	moving-worm graphics demo
caseconvert	converts Modula-2 keywords to uppercase

Forth

Bresehan	circle algorithm example
Analyze	12 templates for the spreadsheet Analyze!

There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Of course, getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12

Executable programs

blink	'slink' compatible linker, but faster, E-D
clean	spins the disk for use with disk cleaners, E-D
epsonset	sends Epson settings to PAR: from menu, E-D
showbig	view hi-res pictures in low-res superbitmap, E-D
speaktime	tell the time, E-D
undelete	undeletes a file, E-D
cnvapi0hm	converts Apple][low, medium and high res pictures to IFF, E-D
menued	menu editor produces C code for menus, E-D
quick	quick disk-to-disk nibble copier, E-D
quickEA	copies Electronic Arts disks, removes protection, E-D
txed 1.3	demo of text editor from Microsmiths, E-D

C programs

spin3	rotating blocks graphics demo, S-E-D
popdi	start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite	VSprite example code from Commodore, S-E-D
AmigaBBS	Amiga Basic bulletin board prog., S-D

Assembler programs

star10	makes star fields like Star Trek intro,S-E-D

Pictures

Mount Mandelbrot	3D view of Mandelbrot set
Star Destroyer	hi-res Star Wars starship
Robot	robot arm grabbing a cylinder

Texts

vendors	Amiga vendors, names, addresses
cardco	fixes to early Cardco memory boards
cinclude	cross-reference to C include files, who includes what
mindwalker	clues to playing the game well
slideshow	make your own slideshows from the Kaleidoscope disk

AMICUS Disk 13

Amiga Basic programs

Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs

gravity	Sci Amer Jan 86 gravitation graphic simulation, S-E-D

Texts

MIDI	make your own MIDI instrument
interface,	with documentation and a hi-res
schematic	picture.

AMICUS Disk 14

Several programs from Amazing Computing issues:

Tools

Dan Kery's	C structure index program, S-E-D

Amiga Basic programs

BMAP Reader	by Tim Jones
IFFBrush2BOB	by Mike Swinger
AutoRequester	example
DOSHelper	Windowed help system for CLI commands, S-E-D
PETrans	translates PET ASCII files to ASCII files, S-E-D
C Squared	Graphics program from Scientific American, Sept86, S-E-D
crlf	adds or removes carriage returns from
files,	S-E-D
dpdecode	decrypts Deluxe Paint, removes copy protection, E-D
queryWB	asks Yes or No from the user, returns
exit	code, S-E
vc	VisiCalc type spreadsheet, no mouse
control,	E-D
view	views text files with window and slider gadget, E-D

Oing, Sproing, yaBoing, Zoing are sprite-based Boing! style demos, S-E-D

CLIClock, sClock, wClock are window border clocks, S-E-D

Texts

An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS 15

The C programs include:

'pr'	a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm'	displays a chart of the blocks allocated on a disk.
'Ask'	questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat'	an enhanced version of AmigaDOS 'status' command.
'Dissolve'	random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2'	invoke new CLI window at the press of a key.

The executable programs include:

'Form'	file formatting program through the printer driver to select print styles
'DiskCat'	catalogs disks, maintains, sorts,merges lists of disk files
'PSound'	SunRize Industries' sampled sound editor & recorder
'Iconmaker'	makes icons for most programs
'Fractals'	draws great fractal seascapes and mountainscapes.
'3D Breakout'	3D glasses, create breakout in a new dimension
'AmigaMonitor'	displays lists of open files, memory use, tasks, devices and ports in use.
'Cosmoroids'	version of 'asteroids' for the Amiga.
'Sizzlers'	high resolution graphics demo written in Modula 2.

Texts:

'ansi.txt'	explains escape sequences the CON: device responds to.
'FKey'	includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn'	programmer's document from
Commodore	Amiga, describes ways to use the
Amiga's	multitasking capabilities in your own programs.

AmigaBasic programs:

'Grids'	draw sound waveforms, and hear them played.
'Light'	a version of the Tron light-cycle video game.
'MigaSol'	a game of solitaire.
'Stats'	program to calculate batting averages
'Money'	"try to grab all the bags of money that you can."

AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS 16

'juggler'	demo by Eric Graham. a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are tipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.

IFF pictures

parodies of the covers of Amiga World and Amazing Computing magazines.

C programs:

'inputhandler'	example of making an input handler.
'FileZap3'	binary file editing program
'ShowPrint'	displays IFF picture, and prints it.
'Gen'	program indexes and retrieves C structures and variables declared in the Amiga include file system.

Executable Programs:

'FixHunk2'	repairs an executable program file for expanded memory
'ms2smus'	converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile'	Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park.

AMICUS 17

Telcommunications disk whuch contains six terminal programs.

"Comm" V1.33	term prog. with Xmodem, WXmodem,
"ATerm" V7.2	term prog. includes Super Kermit
"VT-100"V2.6	Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit"	V4D(050) port of the Unix C-Kermit
"VTek" V2.3.1	Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost"	V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk"	expansion memory necessity
"FixObj"	removes garbage characters from modem received files
"Txt"	filters text files from other systems to be read by the Amiga E.C.
"addmem"	executeable version for use with mem expansion article in AC v2.3
'arc'	file documentation and a basic tutorial on un 'arcing files
"arcre"	for makeing "arc" files E.C.

AMICUS Disk 18

Logo	Amiga version of the popular computer language, with example programs, E-D
Tv'Text	Demo version of the TV'Text character generator
PageSetter	Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow	Resizes any CLI window using only CLI commands, E-D
Life3d	3-D version of Conway's LIFE program, E-D
Defdisk	CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS	Lotus-compatible worksheet that makes calendars
SetKey	Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG	Video pattern generator for aligning monitors, E-D
HP-10C	Hewlett-Packard-like calculator, E-D
SetPrefs	Change the Preferences settings on the fly, in C, S-E-D
StarProbe	Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT	C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat	Like Ing, windows on screen run away from the mouse, E-D
DK	Decays" the CLI window into dust, in Modula 2, S-E-D
DropShadow2	Adds layered shadows to Workbench windows, E-D

Disk 19

This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Livel pictures from the Amazing Stories episode that featured the Amiga.

Solve	Linear equation solver in assembly language, S-E-D
Gadgets	Bryan Catley's AmigaBasic tutorial, S-D
Household	Bryan Catley's AmigaBasic household inventory program, S-D
Waveform	Jim Shields' Waveform Workshop in AmigaBasic, S-D
DiskLib	John Kennan's AmigaBasic disk librarian program, S-D
Subscripts	Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean	C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C	Bob Riemersma's example for making small C programs, S-E-D
COMAL.h	Make C look like COMAL header file, S-D
EmacsKey	Makes Emacs function key definitions by Greg Douglas, S-D
AMon 1.1	Snoop on system resource use, E-D
BTE	Bard's Tale character editor, E-D
Size	CLI program shows the size of a given set of files, E-D
WinSize	CLI window utility resizes current window, S-E-D

Disk 20

Compactor, Decoder	Steve Michel AmigaBasic tools, S-D
BobEd	BOB and sprite editor written in C, S-E-D
SpriteMaster II	Sprite editor and animator by Brad Kiefer, E-D
BlitLab	Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic	Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Benkn	Complete home banking program, balance your checkbook! E-D
cons	Console device demo program with supporting macro routines.
freemap	Creates a visual diagram of free memory
input.dev	sample input handler, traps key or mouse events
joystick	Shows how to set up the gameport device as a joystick.
keyboard	demonstrates direct communications with the keyboard.
layers	Shows use of the layers library
mandelbrot	IFF Mandelbrot program
mouse	hooks up mouse to right joystick port
one.window	console window demo
parallel	Demonstrates access to the parallel port.
printer	opening and using the printer, does a screen dump, not working
print.support	Printer support routines, not working.
proctest	sample process creation code, not working
region	demos split drawing regions
samplefont	sample font with info on creating your own
serial	Demos the serial port
singlePlayfield	Creates 320 x 200 playfield
speechtoy	latest version of cute speech demo
speech.demo	simplified version of speechtoy, with IO requests
text.demo	displays available fonts
timer	demos timer.device use
trackdisk	demos trackdisk driver

Fred Fish Public Domain Software

Fred Fish Disk 1:

amigademo	Graphical benchmark for comparing
amigas.	
amigaterm	simple communications program
with	Xmodem
balls	simulation of the "kinetic thingy"
with balls	on strings
colorful	Shows off use of hold-and-modify
mode.	
dhrystone	Dhrystone benchmark program.
dotty	Source to the "dotty window" demo
on the	Workbench disk.
freedraw	A small "paint" type program with lines, boxes, etc.
gad	John Draper's Gadget tutorial program
gfxmem	Graphical memory usage display prog.
halfbrite	demonstrates "Extra-Half-Brite" mode, if you have it
hello	simple window demo
ieffp	accessing the Motorola Fast Floating Point library from C
palette	Sample prog. to design color palettes.

trackdisk — Demonstrates use of the trackdisk driver.
requesters — John Draper's requester tutorial and example program.
speech — Sample speech demo program.
— Stripped down "speechtoy".
speechtoy — Another speech demo program.

Fred Fish Disk 2:

alib — Object module librarian.
cc — Unix-like frontend for Lattice C compiler.
dbug — Macro based C debugging package. Machine independent.
make — Subset of Unix make command.
make2 — Another make subset command.
microemacs — Small version of emacs editor, with macros, no extensions
portar — Portable file archiver.
xrf — DECUS C cross reference utility.

Fred Fish Disk 3:

gothic — Gothic font banner printer.
roff — A "roff" type text formatter.
ff — A very fast text formatter
cforth — A highly portable forth implementation. Lots of goodies.
xlisp — Xlisp 1.4, not working correctly.

Fred Fish Disk 4:

banner — Prints horizontal banner
bgrep — A Boyer-Moore grep-like utility
bison — GNU Unix replacement 'yacc', not working.
bm — Another Boyer-Moore grep-like utility
grep — DECUS grep
kermit — simple portable Kermit with no connect mode.
MyCLI — Replacement CLI for the Amiga. V. 1.0
mandel — A Mandelbrot set program, by Robert French and RJ Mical

Fred Fish Disk 5:

cons — Console device demo program with supporting macro routines.
freemap — Creates a visual diagram of free memory
input.dev — sample input handler, traps key or mouse events
joystick — Shows how to set up the gameport device as a joystick.
keyboard — demonstrates direct communications with the keyboard.
layers — Shows use of the layers library
mandelbrot — IFF Mandelbrot program
mouse — hooks up mouse to right joystick port
one.window — console window demo
parallel — Demonstrates access to the parallel port.
printer — opening and using the printer, does a screen dump, not working
print.support — Printer support routines, not working.
proctest — sample process creation code, not working
region — demos split drawing regions
samplefont — sample font with info on creating your own
serial — Demos the serial port
singlePlayfield — Creates 320 x 200 playfield
speechtoy — latest version of cute speech demo
speech.demo — simplified version of speechtoy, with IO requests
text.demo — displays available fonts
timer — demos timer.device use
trackdisk — demos trakodisk driver

Fred Fish Disk 6:

compress — like Unix compress, a file squeezer
dadc — analog clock impersonator
microemacs — upgraded version of microemacs from disk 2
mutt — removes multiple occuring lines in files
scales — demos using sound and audio functions
setparallel — Allows changing parallel port parameters
setserial — Allows changing serial port parameters.
sortc — quicksort based sort program, in C
stripc — Strips comments and extra whitespace from C source

Fred Fish Disk 7:

This disk contains the executables of the game Hack V 1.0.1.

Fred Fish Disk 8:

This disk contains the C source to Hack on disk 7.

Fred Fish Disk 9:

moire — Draws moire patterns in black and white
MVP-FORTH — Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems.
proff — a more powerful text formatting program
setiace — Program to toggle interlace mode on and off.
skewb — a rubic's cube type demo
sparks — moving snake Graphics demo

Fred Fish Disk 10:

conquest — An interstellar adventure simulation game
dehex — convert a hex file to binary
filezap — Patch program for any type of file.
fixobj — Strip garbage off Xmodem transferred files.
iff — Routines to read and write iff format files.
ld — simple directory program
ls — Minimal UNIX ls, with Unix-style wildcarding, in C
sq,usq — file squeeze and unsqueeze

trek73 — Star Trek game
yachtc — Dice game.

Fred Fish Disk 11:

dpslide — slide show program for displaying IFF images with miscellaneous pictures

Fred Fish Disk 12:

amiga3d — Shows a rotating 3 dimensional solid "Amiga sign".
ArgoTerm — a terminal emulator program, written in assembler
arrow3d — Shows a rotating 3 dimensional wire frame arrow.
ld4 — directory listing program
IconExec
SetWindow — two programs for launching programs from Workbench that presently only work under CLI
SetAlternate — Makes an icon show a second image when clicked once
StarTerm — terminal emulator, with ASCII Xmodem, dialer, more.

Fred Fish Disk 13:

A Bundle of Basic programs, including:

Jpad	toybox	ezspeak
mandlebrot	xmodem	3dsolids
addbook	algebra	ror
amgseq1	amiga-copy	bend
bounce	box	brickout
canvas	cardl	circle
colorcircles	Copy	cubes1
cutpaste	date	dogstar
dragon	draw	dynamictriangle
Eliza	eztcrm	filibuster
fractal	fscape	gomoku
dart	haiku	hal9000
halley	hauntedM	hidden
join	loz	mandel
menu	minipaint	mouse
Orthello	patch	pena
pinwheel	qbox	random-circles
Readme	rgb	rgbtest
Rord	sabotage	salestalk
shades	shapes	shuttle
sketchpad	spaceart	speakspeach
speecheasy	spell	sphere
spirel	striper	superpad
suprshr	talk	terminal
termtest	tom	topography
triangle	wheels	xenos
xmostriper		

(note: some programs are Abasic, most are Amigabasic, and some programs are presented in both languages)

Fred Fish Disk 14:

amiga3d — update of #12, includes C source to a full hidden surface removal and 3D graphics
beep — Source for a function that generates a beep sound
dex — extracts text from within C source files
dimensions — demonstrates N dimensional graphics
filezap — update of disk 10, a file patch utility
gfxmem — update of disk 1, graphic memory indicator
usage
gi — converts IFF brush files to Image
struct, in C — text.
pdterm — simple ANSI VT100 terminal emulator, in 80 x 25 screen
shell — simple Unix 'csh' style shell
termcap — mostly Unix compatible 'termcap' implementation.

Fred Fish Disk 15:

Blobs — graphics demo, like Unix 'worms'
Clock — simple digital clock program for the title bar
Dazzle — An eight-fold symmetry dazzler program. Really pretty!
Fish — double buffered sequence cycle animation of a fish
Monopoly — A really nice monopoly game written in AbasiC.
OkidataDump — Okidata ML92 driver and WorkBench screen dump program.
Polydraw — A drawing program written in AbasiC.
Polyfractals — A fractal program written in AbasiC.

Fred Fish Disk 16:

A complete copy of the latest developer IFF disk

Fred Fish Disk 17:

The NewTek Digi-View video digitizer HAM demo disk

Fred Fish Disk 18:

AmigaDisplay — dumb terminal program with bell, selectable fonts
Ash — Prerelease C Shell-like shell program, history, loops, etc.
Browser — wanders a file tree, displays files, all with the mouse
MC68010 — docs on upgrading your Amiga to use a MC68010
Multidim — rotate an N dimensional cube with a joystick
PigLatin — SAY command that talks in Pig Latin
Scrimper — Screen image printer
Xlisp1.6 — source, docs, and executable for a Lisp interpreter.

Fred Fish Disk 19:

BlackJack — text-oriented blackjack game
JayMinerSlides — Slides by Jay Miner, Amiga graphics designer, showing flowchart of the internals, in 640 x 400.
chip
Amiga
Keymap_Test — test program to test the keymapping routines
LockMon — Find unclosed file locks, for programs that don't clean up.

Fred Fish Disk 20:

AmigaToAtari — converts Amiga object code to Atari format
DiskSalv — program to recover files from a trashed AmigaDOS disk.
Hash — example of the AmigaDOS disk hashing function
Hd — Hex dump utility a la Computer Language magazine, April 86
MandelBrots — Mandelbrot contest winners
MultiTasking — Tutorial and examples for Exec level multitasking
Pack — strips whitespace from C source
PortHandler — sample Port-Handler program that performs. Shows BCPL environment clues.
Random — Random number generator in assembly, for C or assembler.
SetMouse2 — sets the mouse port to right or left
SpeechTerm — terminal Emulator with speech capabilities, XModem
TxEd — Demo editor from Microsmith's Charlie Heath

Fred Fish Disk 21

This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good!

Fred Fish Disk 22

This disk contains two new "strains" of microemacs.

Lemacs — version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more.
Pemacs — By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys.

Fred Fish Disk 23

Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine.

Fred Fish Disk 24:

Conques — interstaller adventure simulation game
Csh — update to shell on Disk 14, with built in commands,named variables substitution.
Modula-2 — A pre-release version of the single pass Modula-2 compiler originally developed for MacIntosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only.

Fred Fish Disk 25

Graphic Hack — A graphic version of the game on disks 7 and 8 This is the graphics-oriented Hack game by John Toebes. Only the executable is present.

Fred Fish Disk 26

UnHunk — Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specificatio of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM pro grammer. By Eric Black.

C-kermit — Port of the Kermit file transfer program and server.
Ps — Display and set process priorities
Archx — Yet another program for bundling up text files and mailing or posting them as a single file unit.

Fred Fish Disk 27

ABdemos — Amiga Basic demos from Carolyn Scheppner.
NewConvertFD — creates .bmaps from fd files.
BitPlanes — finds addresses of and writes to bitplanes of the screen's bitmap.
AboutBmaps — A tutorial on creation and use of bmaps.
LoadILBM — loads and displays IFF ILBM pics.
LoadACBM — loads and displays ACBM pics.
ScreenPrint — creates a demo screen and dumps it to a graphic printer.
Disassem — Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassembler routines are set up to be callable from a user program so instructions in memory can be disassembled dynamically. By Bill Rogers.
DvorakKeymap — Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns of C-A.
Hypocycloids — Spirograph, from Feb. 84 Byte.
LinesDemo — Example of proportional gadgets to scroll a SuperBitMap.
MemExpansion — Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Fellinger.
SafeMalloc — Program to debug 'malloc()' calls
ScienceDemos — Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galileen satellite plotter. By David Eagle.

Fred Fish Disk 28

ABasic games by David Addison: Backgammon , Cribbage, Milestone, and Othello
Cpp — DECUS 'cpp' C preprocessor, and a modified 'cc' that knows about the 'cpp', for Manx C.

Shar — Unix-compatible shell archiver, for packing files for travel.
SuperBitMap — Example of using a ScrollLayer, syncing SuperBitMaps for printing, and creating dummy RastPorts.

Fred Fish Disk 29

AegisDraw Demo — Demo program without save and no docs.
Animator Demo — Player for the Aegis Animator files
Cc — Unix-like front-end for Manx C
Enough — Tests for existence of system resources, files, and devices
Rubik — Animated Rubik's cube program
StringLib — Public domain Unix string library functions
Vt100 and — VT-100 terminal emulator with Kermit Xmodem protocols

Fred Fish Disk 30

Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.

BBS — an Amiga Basic BBS by Ewan Grantham
FineArt — Amiga art
FontEditor — edit fonts, by Tim Robinson
MenuEditor — Create menus, save them as C source, by David Pehrson
StarTerm3.0 — Very nice telecommunications by Jim Nangano

(Fred Fish Disk#30 is free when ordered with at least three other disks from the collection.)

Fred Fish Disk 31

Life — Life game, uses bitter to do 19.8 generations a second.
Mandelbrot — Version 3.0 of Robert French's program.
MxExample — Mutual exclusion gadget example.
RamSpeed — Measure relative RAM speed, chip and fast.
Set — Replacement for the Manx "set" command for environment variables, with improvements.
Tree — Draws a recursive tree, green leafy type, not files.
TxEd — Crippled demo version of Microsmith's text editor, TxEd.
VDraw — Full-featured drawing program by Stephen Vermeulen.
Xicon — Invokes CLI scripts from icon
Ticon — Displays text files from an icon.

Fred Fish Disk 32

Address — Extended address book written in AmigaBasic.
Calendar — Calendar/diary program written in AmigaBasic.
DosPlus1 — First volume of CLI oriented tools for developers.
DosPlus2 — Second volume of CLI oriented tools for developers.

Executables only:

MacView — Views MacPaint pictures in Amiga low or high res, no sample pictures, by Scott Evernden.
Puzzle — Simulation of puzzle with moving square tiles.
ShowHAM — View HAM pictures from CLI
Solitaire — ABasiC games of Canfield and Klondike, from David Addison.
Spin3 — Graphics demo of spinning cubes, double-buffered example.
Sword — Sword of Fallen Angel text adventure game written in Amiga Basic.
Trails — Leaves a trail behind mouse, in Modula-2

Fred Fish Disk 33

3dstars — 3d version of the "stars" program below.
Bigmap — Low-level graphics example scrolls bitmap with ScrollVPort.
Dbuf.gels — Double-buffered animation example for BOBs and VSprites.
DiskMapper — Displays sector allocation of floppy disks.
MemView — View memory in real time, move with joystick.
Cing — Bouncing balls demo
Sproing — Cing, with sound effects.
ScreenDump — Dumps highest screen or window to the printer.
Sdb — Simple database program from a DECUS tape.
Stars — Star field demo, like Star Trek.
TermPlus — Terminal program with capture, library, function keys, Xmodem, CIS-B protocols.
Vt100 — Version 2.0 of Dave Wecker's VT-100 emulator, with scripts and function

Fred Fish Disk 34

Alint — Support files for Gimpel's 'lint' syntax checker
Blink — PD 'alink' compatible linker, faster, better.
Browser — Updated to FF 18 'browser', in Manx, with scroll bars, bug fixes.
Btree — b-tree data structure examples
Btree2 — Another version of 'btree'
Calendar — Appointment calendar with alarm.
Less — File viewer, searching, position by percent, line number.
NewFonts — Set of 28 new Amiga fonts from Bill Fischer
Pr — Background print utility, style options, wildcards.
Requester — Deluxe Paint-type file requester, with sample.

Fred Fish Fred Fish Disk 35

ASendPacket	C example of making asynchronous I/O calls to a DOS handler, written by C-A
ConsoleWindow	C example of getting the Intuition pointer a CON: or RAW: window, for tidy by C-A.
DirUtil	Walk the directory tree, do CLI operations from menus
DirUtil2	Another variant of Dirutil.
FileRequester	Lattice C file requester module, with demo driver, from Charlie Heath.
MacView	Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.
Plop	Simple IFF reader program
PopCLI	Sidekick-style program invokes a new CLI, with automatic screen blanking.
QuickCopy	Devenport disk copiers duplicate copy-protected disks.
ScrollPf	Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.
SendPacket	General purpose subroutine to send AmigaDos packets.
SpriteMaker	Sprite editor, can save work as C data structure. Shareware by Ray Larson.
Tracker	Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.
TriClops 3-D	space invasion game, formerly commercial, now public domain. From Geodesic Publications.
Tsize	Print total size of all files in subdirectories.
Unifdef	C preprocessor to remove given #ifdef'd sections of a file, leaving the rest alone. By Dave Yost.
Vttest	VT-100 emulation test program. Requires a Unix system.

Fred Fish Disk 36

Acp	Unix-like 'cp' copy program
Clock	Updated version of clock on disk 15.
Csh	Manx 'csh'-like CLI, history, variables, etc.
DietAid	Diet planning aid organizes recipes, calories
Echo	Improved 'echo' command with color, cursor addressing
FixHunk	Fixes programs to let them run in external memory.
Fm	Maps the sectors a file uses on the disk.
KickBench	Docs, program to make a single disk that works like a Kickstart and Workbench.
Lex	Computes Fog, Fleech, and Kincaid readability of text files.
TunnelVision	David Addison ABasic 3D maze perspective game.
Vc	Visicalc-like spreadsheet calculator program.
Vt100	Version 2.2 of Dave Wecker's telecom program
YaBoing	Cingl style game program shows sprite collision detects

Fred Fish Disk 37

This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersley at Washington State University.

Fred Fish Disk 38

CSquared	Sep 86 Sci American, Circle Squared algorithm
FixObj	Strips garbage off Xmodem transfered object files
Handler	AmigaDOS handler (device) example from C-A
Hp-10c	Mimics a HP-10C calculator, written in Modula-2
IFFEncode	Saves the screen as an IFF file
IffDump	Dumps info about an IFF file
Jsh	BDS C-like CLI shell
NewStat	STATUS-like program, shows priority, processes
Reversi	Game of Reversi, version 6.1
UUdecode	Translate binary files to text, Unix-like programs
Vdraw	Drawing program, version 1.14
VoiceFiter	DX MIDI synthesizer voice filer program
Window	Example of creating a DOS window on a custom screen

Fred Fish Disk 39

AnsiEcho	'echo', 'touch', 'list', 'cls' written in assembler.
Display	Displays HAM images from a ray-tracing program, with example pictures.
Driver	Example device driver source, acts like RAM: disk
Xlisp	XLisp 1.7, executable only

Fred Fish Disk 40

Ahost	Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.
AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.
Arc	Popular file compression system, the standard for transiting files
AreaCode	Program that decodes area codes into state and locality.
Blink	'alink' replacement linker, version 6.5
Cosmo	An 'asteriods' clone.
Dg210	Data General D-210 Terminal emulator
DirUtil	Windowed DOS interface program, version 1.4
DOSHelper	Windowed AmigaDOS CLI help program
PagePrint	Prints text files with headers, page breaks, line numbers
PopCLI	Starts a new CLI with a single keystroke, from any program, With a screen-saver feature. Version 2, with source.
SpriteEd	Sprite Editor edits two sprites at a time
X-Spell	Spelling checker allows edits to files

FF 41

AmigaVenture	Create your own text adventure programs in AmigaBasic.
Csh	Version 2.03 of Dillon's Csh-like shell. Executable only
Dbug	Macro based C debugging package, update to FF #2
DualPlayField	example from CBM, update to Intuition manual
GetFile	Heath's file requester, with source
LstXref	Cross reference of Lattice 3.10 header files
Lines	Line drawing demo program
SetFont	Changes font used in a CLI window
Vt100	Version 2.3 of the VT-100 terminal program.

FF 42

This disk contains an Amiga version of MicroGNUEmacs.

FF 43

BasicBoing	AmigaBasic program demos page tipping of a 3D cube
Bbm	Demo copy of B.E.S.T. Business Management System.
BbsList	A list of Amiga Bulletin Board Systems
Cc	C compiler frontends for Manx and Lattice C
Copper	A hardware copper list disassembler
InstIFF	Converts Instruments demo sounds to IFF sampled sounds
PopColours	Adjust RGB colors of any screen
SpriteClock	Simple clock is displayed on a sprite above all screens
ST Emulator	Non-serious Atari ST emulator
WBrun	Lets Workbench programs be run from the CLI
Wild	Two Unix shell style wildcard matching routines

FF 44

Icons	Miscellaneous icons
NewIFF	New IFF material from CBM for sampled voice and music files
RayTracePics	The famous ray-tracing pictures, from FF#39, now converted to IFF for *much* faster viewing.
HAM format	
ViewILBM	Displays normal and HAM ILBM files

FF 45

Clue	Clue board game
Make	Another 'make', with more features
Pictures	Miscellaneous pictures
Update	Updates an older disk with newer files from another disk
Whereis	Searches a disk for files of given name

FF 46

Asm	Shareware 68010 macro assembler,
ROM	Kernel Manual compatible
CheckModem	'execute' file program detects presence of modem
Egad	Gadget editor from the Programmers Network
Jive	Transforms a file from English to Jive.
My.lib	A binary only copy of Matt's alternate runtime library. Author: Matt Dillon
ProffMacros	Subset Berkeley 'ms' and 'mm' macros for 'proff'
ValSpeak	Transforms a file from English to Valley Speak.

FF 47

3D-Arm	Simulation of a robotic arm, very good graphics, teaching tool, including C source.
Juggler	Eric Graham's stunning HAM animation of a robot juggler
VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols

FF 48

Bru	Alpha version of a hard disk file archiver
Comm	Version 1.30 of a terminal emulator with phone directories
Csh	Version 2.04 of Matt Dillon's Unix 'csh'-like CLI replacement, including Lattice and Manx C source
Diskperf	Disk benchmark program for Unix and Amiga
Du	Computes disk storage of a file or directory
MemWatch	Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery.
Profiler	A realtime execution profiler for Manx C programs. Includes C source.

FF 49

Cycloids	Update of electronic spirograph from disk 27
DirUtil	Enhanced version of DirUtil from disk 35
MultiDef	Scans a set of object modules and libraries searching for multiply defined symbols
MyUpdate	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process
Plot	Computes and displays 3 dimensional functions in hires
Polygon	Moire type pattern generator with color cycling
QMouse	Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed.
Touch	Example of setting the datestamp on a file,using a new technique from Commodore-Amiga
Trees	More extensive version of the trees program on Disk 31

FF 50

Asm	Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mnemonics.
BreakOut	A brick breakout game, uses 3-D glasses
DiskZap	Version 1.1 of a program to edit disks and binary files
FirstSilicon	A smart CLI replacement with full editing and recall of previous commands
Missile	A Missile Command-type game, with sound, in assembler
PerfectSound	Sound editor for a low-cost sound digitizer
Sizzlers	Graphics demos
UnixArc	Version of 'arc' for Unix System V machines, in C
Wombat	Version 3.01 of Dave Walker's terminal emulator

FF 51

Bison	GNU for Unix 'yacc', working update to disk 4 version
Compress	Update to the file compression program on Disk 6
Cos	"Wheel of Fortune"-type game in AmigaBasic
DifSeed	Unix-like 'diff' and 'sseq' for finding the differences between two files, and then recreating the other, given one file, and the list of differences.
Sq, Usq	Portable versions of the CP/M squeeze and unsqueeze

FF 52

Assign	Replacement for AmigaDOS 'assign' command in C
Fractal	Makes random fractal terrains
Poly, HAMPoly	Workbench-type demos for making polygons in lores and HAM
MixGads	Example of mutual exclusion gadgets with GadgetText
Tek4010	Tektronix 4010 terminal emulator
VDraw	Versions 1.16 and 1.19 of a Deluxe Paint-like drawing program

FF 53

Animations	Demo animations with player program for Aegis Animator
ARCre	Creates rename scripts for files with long names, so they can be easily 'arc'ed and un'arc'ed.
ARP	Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'filenote' and 'makedir'
Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It will produce simple assembly language output, but needs a lot of work.
Spreadsheet	Update with source of the 'vc' spreadsheet on disk 36
TarSplit	Port of program to split Unix 'tar' archives
UUencode	Utilities to encode and decode binary files for ASCII transmission, expanding them by 35 percent

FF 54

Hanoi	Solves Towers of Hanoi Problem in it's own Workbench window. by Ali Ozer
ISpell	Port of a Unix screen oriented, interactive spelling checker. (Expension RAM required) by Pace Willisson
Ing	A Screen of lots of bouncing little windows by Leo 'Bols Ewhac' Schwab
Lav	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by Willaim Ruckidge
MIDITools	Programs to play/record through the MIDI I/F. by Fred Cassirer
MoreRows	Program to make the WorkBench
Screen	larger than normal. by Neil Katin and Jim Mackraz
Tilt	Program to make your Amiga look like it didn't pass vibration testing. by Leo Bols Ewhac' Schwab

FF 55

Csh	V2.05 of Matt Dillon's csh like shell (Modified for Manx C). by Matt Dillon, Modified by Steve Drew
NewStartups	New C Startup modules:
AStartup.asm	with 1.2 fixes and better quote handling.
TWStartup.asm	opens a stdio window, using user specs. by Commodore, posted to BIX by Carolyn Schepper
Palette	Change another program's screen colors. by Carolyn Schepper
PipeDevice	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon
ScreenSave	Save a normal or HAM mode screen as an IFF file. by Carolyn Schepper
ShanghaiDemo	Demo version of the Activision game Shanghai.
SoundExample	A double buffered sound example for Manx C. by Jim Goodnow
Vsprites	A working vsprite example. by Eric Cotton
Vt100	V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem. by Dave Wecker

FF 56

ClipBoard	Clipboard device interface routines, to provide a standard interface. by Andy Finkle
ConPackets	Demos the use of DOS Packets, ConUnit, etc. by Carolyn Schepper
GetDisks	Program to find all available disk device names and return them as an exec list. by Philip Lindsay
GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck McManis
Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Schepper
MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Schepper
mCAD	An object oriented drawing program, V1.1 by Tim Mooney

FF 57

CutAndPaste	Implementaions of Unix cut and paste commands. by John Weald
GraphIt	Program to plot simple functions in 2 or 3 dimensions. by Flynn Fishman
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham
MouseReader	Shareware program to read text files and view IFF files using only the mouse. by William Betz
Ogre	Game of tactical ground combat in the year 2086. by Michael Caplinger; Amiga port by Hobie Orris
Splines	Program to demonstrate curve fitting and rendering techniques. by Helene (Lee) Taran

FF 58

ASDG-rd	Extremely useful shareware recoverable ram disk. by Perry Kivolowitz
BigView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson
EGraph	Reads pairs of x and y value from a list of files and draws a formatted graph. by Laurence Turner
HyperBase	Shareware data management system. V1.5
MemClear	Walks through the free memory lists, zeroing free memory along the way. by John Hodgson
NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgeson
RainBow	A Meurauder-Style rainbow generator. by John Hodgson
SMUSPlayers	Two SMUS plays, to play SMUS IFF music formatted files. by John Hodgson
View	A tiny ILBM viewer by John Hodgson
WBdump	JX-80 optimized workbench printer that does not use DumpRPort. by John Hodgson

ed Fish Disk 59

Browser	Update to browser program on disks
18 and 34. S-E	
Browser2	Another different browser program.
E	
Clock	Clock program with fonts, colors. E
Dme	Dillon text editor V1.22 for
programmers. E-D	
DropCloth	Puts a pattern on the Workbench
backdrop. E-D	
DropShadow	Puts shadows on Workbench windows.
E-D	
FixWB	Similar to DropCloth, but doesn't work
yet. S-D	
mCAD	Object-oriented drawing program,
version 1.2.2. Much improved over disk 56.	
Robotroff	Demo of animated pointers on
Workbench. S-E-D	
Supermort	General compounding/amortization loan
calculator. E-D	

Fred Fish Disk 60

Various shareware and freeware programs

Blitz	Memory resident file viewer. Very
fast. E-D	
BlitzFonts	Makes text output faster. E-D
HandShake	Terminal emulator with
VT52/VT100/VT102support. E-D	
Med	Mouse-driven text editor version 2.1.
E-D	
PrtDrvGen	Generates printer drivers, version 1.1.
Source available from author. E-D	
Show	Slideshow-like IFF viewer, V2.1. E-D
Uedit	Customizable text editor V2.0. E-D
Ueturbo	Example Uedit setup macros. S-E-D

Fred Fish Disk 61

ATPatch	Patches Transformer to work under AmigaDOS 1.2. S-E-D
FillDisk	Writes zeroes to free blocks on a disk for security. S-E-D
LPatch	Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D
MicroEmacs	Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D
PearlFont	Like Topaz, but rounded edges.
Terrain	Generates fractal scenery. S-E-D
VSprites	Makes 28 Vsprites, from Peck book. S-E-D

Fred Fish Disk 62

This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.

Fred Fish Disk 63

This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.

Fred Fish Disk 64

This is an offical IFF specification disk from Commodore, an update to disk 16.

Fred Fish Disk 65

Bawk	Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D.
MWB	Example of rerouting Workbench window open calls to another
custom	screen. Version 1.01, S-E-D
CloseWB	Example for closing a custom Workbench screen. S-E-D
Cookie	Generates one-line fortune-cookie aphorisms. S-E-D
JTime	Build-your-own mouse port clock.
MenuBuilder	Creates C source files for menus, based on text descriptions. S-E-D.
NewPackets	CBM tutorial on new packets and structures in AmigaDos 1.2.
PascalToC	Pascal to C translator, not so great. S-E-D
Prep	'ratfor'-like FORTRAN preprocessor. S-E-D
RunBack	Starts programs from CLI, allowing CLI window to close. E-D
SunMouse	This program automatically clicks in windows when the mouse is moved over them. Version 1.0, E-D

Fred Fish Disk 66

AmScsi	Preliminary plans for a SCSI disk controller board.
Asm68k	Macro assembler, version 1.0.1. E-D
Assigned	Example for avoiding DOS insert-disk requester, by scanning the list of 'assigned names. S-E-D
Dk	Pretends to eat away at CLI window. S-E-D
Flip	Flips whole screen as a joke. S-E-D
Foogol	Foogol cross-compiler generates VAX assembly code. S-E-D
Free	Prints amount of free space on all drives. S-E-D
MallocTest	malloc/free memory test program. S-E-D
Melt	Pretends to melt the screen. S-E-D
Nart	Graphic flying string demo. S-E-D
Purty	Easy way to set printer attributes from Workbench. E-D
RayTracer	Simple ray tracing program. E-D
SendPackets	Updated CBM examples of packet routines on disk 35. S-E-D
SnapShot	Memory resident screen dump. E-D
TagBBS	Shareware BBS system, version 1.02.

Fred Fish Disk 67

AmCat	Shareware disk cataloging program. E-D
AmigaSpell	Shareware Intuition spelling checker, V2.0. E-D
Bouncer	3-D bouncing ball written in MuttiForth, S-E-D
Comm	Terminal program version 1.33, E
Dux5	Another version of DirUtil. S-E-D
HexCalc	Hex, octal, & decimal calculator. E-D
Icons	Various big and alternate image icons.
Mandala	Mandala graphics and sound. E
PersMail	Demo shareware personal file manager.
RSLClock	Menu bar clock version 1.3. E-D
RTCubes	Graphics demo of 3D cubes. E-D
Wheel	"Wheel of Fortune"-type game, in AmigaBasic

Fred Fish Disk 68

This is version MG 1b of the MicroGNUEmacs. Source and executable areincluded, as well as source for other computers besides the Amiga.

Fred Fish 69

Asm68k	Macro assembler, v1.0.3, E-D
BlitLab	Blitter exploring program, in C, S-E-D
Conman	Replacement console device handler adds editing and history to any application that uses CON:, v0.9, E-D
Console	Replacement console routines, in C, S-E-D
Dk	Decays the screen bit by bit, update to disk 66, in Module-2, S-E-D
Frags	Displays memory fragmentation by listing the size of free memory blocks, in C, S-E-D
IconType	Change the type of an icon, in C, S-E-D
Make	'make' in Manx C, S-E-D
MonProc	Monitors processes for packet activity, in C, S-E-D
MouseClock	Turns mouse pointer into a digital clock, in C, S-E-D
Sb	Browses system structures, from Transactor magazine, v1.0, in C, S-E-D
Spew	Generates 'National Enquirer'-type headlines from rules file. In C,S-E-D
Spool	Three programs to demonstrate multitasking and spooling in a printer spooler. In C, v1.2, S-E-D
Wc	Counts words ala Unix 'wc', but faster, in C, S-E-D

Fred Fish 70

This is a disk of shareware programs.

AmigaMonitor	Explores state of the system, v1.13
Arc	Standard file compressor andl librarian, v0.23, a port of MS-DOS v5.0. E-D
BlackBook	Phone book program.
DoTil	Intuition-driven file manipulator program, v2.0.
GravityWars	Game of planets, ships and black holes, v1.03.
Jobs	Alternate user interface to CLI and Workbench, v2.1.
Lens	Magnifies area around mouse, shows it in a window, v1.0.
Life-3d	3D version of the classic cellular-automaton game, v1.2.
Logo	Logo language interpreter
SetKey	Demo keymap editor, v1.0
Vpg	Makes displays for aligning video monitors, v1.0.

Fred Fish 71

AirFoil	Makes airfoils using the Joukowski transformation, in C, S-E-D
Amiga Basic	Miscellaneous programs including: 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D
Blocks	A variation of "lines", but with variable color blocks. E-D
Comm	Great terminal program, v1.34, E-D
DiskX	Utility for exploring file system.E-D
Fpic	Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D
IconMk	Makes icons for files that don't have them, v1.2a, E-D
Icons	New icons
NewFonts	Two new fonts; 'shalt18', an electronic cirult element font, and 'ibm5', a PC-like font.
PetCLI	An AmigaBASIC CLI shell program.
PWDemo	Demo of the commercial product PowerWindows, v1.2. It aids creation of custom windows, menus, and gadgets, giving C or assembly source. E-D
Rot	Creates and animates 3-D objects, v0.5, E-D
TimeSet	Sets time from Workbench, E-D

Fred Fish 72

This is a disk of IFF pictures.

Fred Fish 73

Add	Customizes existing program menus with Amiga-key shortcuts. Also includes 'until', which waits until a given window is created. Shareware, in C, S-E-D.
AutoIconOpen	Fools WB into thinking mouse has double-clicked icons. In C, S-E-D
Dio	Generic Exec device interface code for opening libraries, getting multiple I/O channels,asynchronous operations, etc. In C, S-E-D.
Dissolve	Slowly displays IFF files, ala Nov 86 Dr. Dobb's program. In C, S-E-D
DTerm	Flexible, reprogrammable terminal program v1.10, E-D
Expose	Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D.
Lit	Scans a text file, converts to C-style printable strings.C,v2.0, S-E-D
Lmv	"Long Movie", program views series of IFF picts in quick succession, up to 19 fps. Shareware, E-D
MouseOff	Mouse pointer disappears after ten seconds of non-use. In C, S-E-D
PerOut	Examples of controlling parallel port with resources instead of the PAR: device. In C, S-E-D
PenPalFont	Child-like font.
RunBackGround	Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C,S-E-D
SnapShot	Screendump utility,update FF 66.E-D
TypeAndTell	Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D
Xplor	Prints info about system lists, in assembler, S-E-D

Fred Fish 74

Cled	Edits and recalls CLI commands, v1.3, E-D
Control	Intercepts graphic printer dump calls and accesses color map, width,and screen resolution. C,S-E-D
Dme	Simple WYSIWYG text editor for programmers,v1.25. Update of FF 59.E-D
DropShadow	Workbench dropshadows, v2.0. Update to disk 59. E-D
Funds	AmigaBASIC program tracks mutual or stock p-D
Less	Text viewing program, like Unix "more", v1.1, update to disk 34. S-E-D
Makemake	Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D
mCAD	Object-oriented drawing prog. v1.2.4, update to FF 59.Shareware, E-D
Random	Simple random number generator in C. S-E-D
TDebug	Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D
Units	Converts measurements in different units, includes "chart" option, in C, S-E-D
XCopy	Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D

To Be Continued.....

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.

-AC-

Index of Advertisers

Support the Amiga™ & Amazing Computing™,

Write!

Your thoughts, experiences, and programs are needed by others. For an Author's guide, write to:

Author's Guide,
PiM Publications, Inc.,
P.O.Box 869, Fall River,
MA. 02722.

TeleGames is what you've waited for. The Future is here.

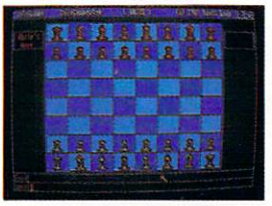
TeleGames allows you to use your computer and modem to play Chess, Checkers and Backgammon with a human opponent over the telephone. Only $34.95!

TeleGames Features

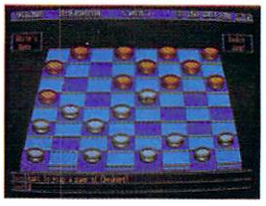
* Chess * Checkers * Backgammon
* Superb Graphic Game Simulations
* Smooth Depth Arranged Movement
* 4 angle 3D & 2D view perspectives
* Digitized Sound Effects
* Compatible with any modem
* 300, 1200, 2400, 9600 Baud
* Call originate or answer
* Null Modem Connect option
* Save Game & Transmit Game options
* Opponent File Directories
* Send and Receive Typed Messages
* Easy to Use Menus & Requesters
* All Official Game Rules Supported
* Play Over the Phone or at Home
* Legal Moves Graphically enacted on the TeleConnected computer
* Fully copyable to hard disks
* Upgrades available on our BBS

3D Chess

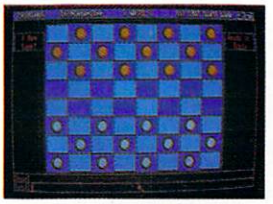
2D Chess



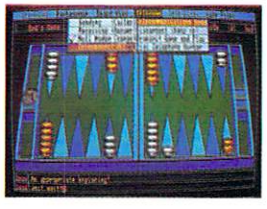
3D Checkers



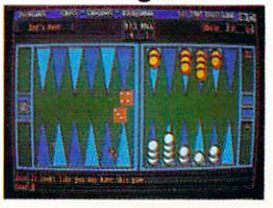
2D Checkers



3D Backgammon



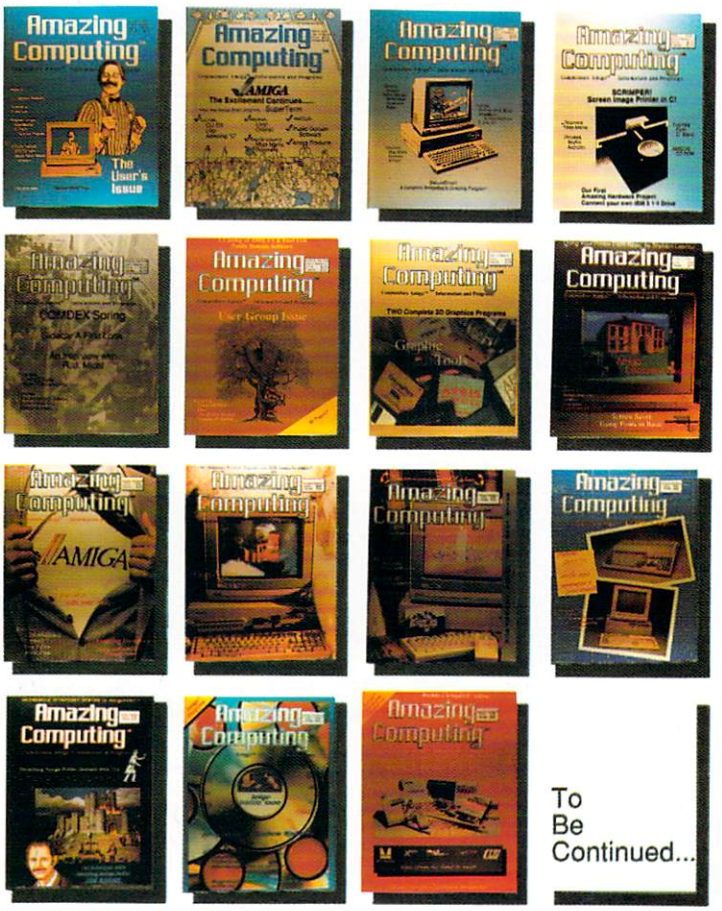
2D Backgammon



If you Enjoy Telecomputing, You'll Love TeleGames!

Published by Software Terminal
3014 Alta Mere, Fort Worth, TX 76116
817-244-4150 Modem: 817-244-4151
Dealer Inquiries Invited

With a past like this....



The future is a tradition.

Since February 1986, **Amazing Computing**™ has been providing users with complete information for their Amigas. With the growth of the new Amiga 500 and Amiga 2000, the Amiga user needs more information than ever in selecting the right software and hardware for their needs.

Amazing Computing™ will continue to offer the Amiga user the best in technical knowledge and in unbiased reviews for the Commodore-Amiga™.

Amazing Computing™ will not rest on past achievements.

To Subscribe to Amazing Computing™ or to purchase Public Domain Software, please fill out the form below and send with a Check or Money Order to:

PiM Publications Inc.
P.O.Box 869
Fall River, MA 02722

Back issues are still available at $4.00 each (Foreign orders, please add $1.00 U.S. per issue for P. & H).
All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.

Amaze Me!

Please start my subscription to Amazing Computing™ with the next available issue. I have enclosed $24.00 for 12 issues in the U.S. ($30.00 Canada and Mexico, $35.00 overseas). All funds must be in U.S. Currency drawn on a U.S. Bank.

Please circle your selection:
Subscription PDS (as noted) Back Issues Sub Renewal

Name _____

Street _____

City _____ **St.** ____ **ZIP** _____

Amount enclosed _____

Mass. Residents, please add 5% sales tax on PDS orders

BACK ISSUES: $4.00 each (foreign orders add $1.00 each for Postage and Handeling

VOL.1#1 VOL.1#2 VOL.1#3 VOL.1#4 VOL.1#5 VOL.1#6
VOL.1#7 VOL.1#8 VOL.1#9 VOL.2#1 VOL.2#2 VOL.2#3
VOL.2#4 VOL.2#5 VOL.2#6

AC 7 87

Public Domain Software:
$6.00 each for subscribers (yes, even new ones!)
$7.00 each for non subscribers.

AMICUS: A1 A2 A3 A4 A5 A6 A7 A8 A9 A10
A11 A12 A13 A14 A15 A16 A17 A18 A19 A20

Fred Fish:
FF1 FF2 FF3 FF4 FF5 FF6 FF7 FF8 FF9 FF10
FF11 FF12 FF13 FF14 FF15 FF16 FF17 FF18 FF19 FF20
FF21 FF22 FF23 FF24 FF25 FF26 FF27 FF28 FF29 FF30
FF31 FF32 FF33 FF34 FF35 FF36 FF37 FF38 FF39 FF40
FF41 FF42 FF43 FF44 FF45 FF46 FF47 FF48 FF49 FF50
FF51 FF52 FF53 FF54 FF55 FF56 FF57 FF58 FF59 FF60
FF61 FF62 FF63 FF64 FF65 FF66 FF67 FF68 FF69 FF70
FF71 FF72 FF73 FF74 FF75 FF76 FF77 FF78 FF79 FF80

LAND OF LEGENDS

Mystery and excitement awaits your every move as you sojourn through one adventure after another in Land of Legends. Truly the most thrilling of all Dungeon & Dragon type adventures.

Participate in the adventures of three brothers as they travel through a faery land full of vicious monsters, enchanted princesses, fearsome dragons and even a kindly wood cutter or two. Faery Tale Adventure will provide you with weeks of challenging exploration to conquer, so pick up your copy and begin today!

ONE-ON-ONE SERIES™

These three arcade style games are the first in our new ONE-ON-ONE SERIES. These exciting games feature one-on-one playability, digitized sounds & music, realistic graphics and fast action. Our reasonably priced ONE-ON-ONE SERIES will keep you on the edge of your seat, so pick up all three today!

OTHER PRODUCTS FROM MICROILLUSIONS

- **DISCOVERY™**
  Now available in: Math version, Spelling version, Trivia version
- **THE PLANETARIUM™**
  Available soon
- **DUNGEON CONSTRUCTION™**
  The second in the "Quest Master Series" will soon be available

All of these products are now being developed for the Amiga, and will soon be available in other formats (C-64/128, IBM/PC, Apple, Atari ST).

microillusions™

7408 Chatsworth St., Granada Hills, CA 91344, inside CA 818/360-3715 • outside CA 800/522-2041 • FAX 818/360-1464

There's a slim difference between the ordinary and the extraordinary . . . PAL JR.

True DMA controller with SCSI option

1MByte FAST RAM at C00000

User expandable to 9MBytes fast RAM

No wait state memory

Battery backed clock/calendar

Open ZORRO expansion slot

Entire system auto-configures

Quiet fan for cooling

20 MByte PAL JR.	$1,495
40 MByte PAL JR.	$1,995
80 MByte PAL JR.	CALL
2/8 MByte Garganturam board	$ 835

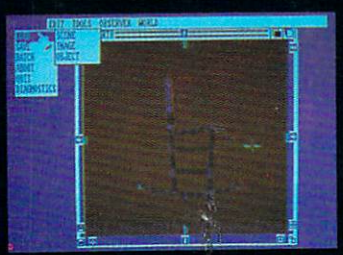
"The PAL JR. is clearly the highest performance Amiga hard disk."
John Foust
Amazing Computing

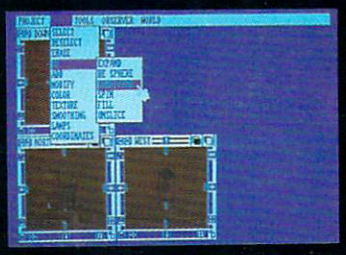
"Designed for the power user . . . A solid well built piece of hardware."
Bruce Webster
Byte Magazine

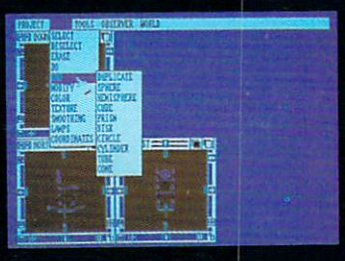
"The PAL System is extremely well built and is the fastest hard disk system available for the Amiga. It's also remarkably easy to install."
Louis Wallace
AmigaWorld Contributing Editor

Create your own universe with SCULPT 3-D

- Brings the power of RAY TRACING to the Amiga
- Full Intuition interface
- Five IFF modes including HAM
- Create images in TRUE 3-D
- Supports overscan display for full screen video
- Powerful as packages costing thousands of dollars more
- Written by Eric Graham a la "Robot Juggler"
- Suggested Retail only $99.95







Watch for ANIMATE 3-D later this summer

BYTE by BYTE™

Arboretum Plaza II 9442 Capital of Texas Highway North Suite 150 Austin, TX 78759 (512) 343-4357